Incremental Object Detection with Prompt-based Methods

Matthias Neuwirth–Trapp[1,2] Maarten Bieshaar[2] Danda Pani Paudel[3] Luc Van Gool[3]
mneuwirth@ethz.ch

[1]ETH Zürich [2]Bosch Research [3]INSAIT, Sofia University "St. Kliment Ohridski"

Abstract

Visual prompt-based methods have seen growing interest in incremental learning (IL) for image classification. These approaches learn additional embedding vectors while keeping the model frozen, making them efficient to train. However, no prior work has applied such methods to incremental object detection (IOD), leaving their generalizability unclear. In this paper, we analyze three different prompt-based methods under a complex domain-incremental learning setting. We additionally provide a wide range of reference baselines for comparison. Empirically, we show that the prompt-based approaches we tested underperform in this setting. However, a strong yet practical method—combining visual prompts with replaying a small portion of previous data—achieves the best results. Together with additional experiments on prompt length and initialization, our findings offer valuable insights for advancing prompt-based IL in IOD.

1. Introduction

In incremental learning (IL), models are sequentially trained on new tasks [46]. This work addresses domain incremental learning (DIL) for object detection, wherein each new task introduces data from a previously unseen domain, though target classes remain consistent across tasks [32, 46]. When training on a new domain, the optimization process updates model weights to minimize task-specific losses, inadvertently overwriting previously learned representations. This phenomenon, termed *catastrophic forgetting*, remains a central challenge in IL. Attempts to mitigate forgetting often reduce model adaptability to new tasks, resulting in the *stability-plasticity dilemma* [25].

Various strategies have been proposed to manage this dilemma, with one promising direction involving learnable prompts [60]. In prompt-based methods, trainable prompts are prepended to inputs of pre-trained transformers to guide task-specific feature extraction [48, 49, 60]. Typically, distinct prompts are allocated per task, and during inference, the appropriate prompt is selected based on task identification [47–49]. Visual prompts differ from textual ones, as they do not convey language meaning and can be considered *pseudo-words*. Despite numerous variations, prompt-based methods have mostly been evaluated only on classification tasks, leaving their effectiveness in other computer vision objectives largely unexplored [32, 60].

In this paper, we present the first study of prompt-based IL methods applied to DIL for object detection. We establish several robust baselines and systematically evaluate three widely-used prompt-based IL methods—L2P [49], DualPrompt [48], and S-Prompt [47]—under varying configurations. We extend our analysis by examining prompt length and prompt initialization strategies.

Our experiments leverage the challenging D-RICO benchmark [36], consisting of 15 tasks from automotive and surveillance domains—key application areas for object detection. D-RICO integrates data from 14 diverse datasets, spanning imaging sensors, lens types, perspectives, environmental conditions (*e.g.*, weather, daytime), and both synthetic and real-world scenarios. This benchmark thus embodies significant distributional shifts, offering a rigorous framework for evaluating IL methods.

We demonstrate that although the three examined prompt-based methods perform well on classification tasks, they significantly underperform on object detection within D-RICO. Combining these findings with detailed analyses of prompt initialization strategies and optimal prompt lengths, we provide a comprehensive understanding of the factors influencing prompt-based IL performance, thereby paving the way for future developments in this area.

Our main contributions are:

- We are the first to study prompt-based IL for object detection, showing common methods underperform, with DualPrompt as the most effective tested method.
- Our work presents strong baselines and shows that combining visual prompt tuning with replaying previous task data is a practical and straightforward approach to IL.
- Further investigations show that choosing a fixed prompt length is sufficient across tasks, and initializing prompts with lower values is more effective.

2. Related Works

2.1. Incremental Object Detection

Object detection models broadly fall into two categories: single-stage detectors, which focus on fast inference [4, 18], and two-stage detectors, known for their higher accuracy [3, 10, 12, 38]. In incremental learning (IL) for object detection, two-stage models have traditionally dominated research [32, 34, 42, 44], though incremental learning with single-stage is increasingly explored [8, 29, 33, 40]. To mitigate catastrophic forgetting, distillation-based regularization techniques [7–9, 14, 24, 33, 34, 37], as well as rehearsal methods that replay previously seen data [26, 28, 35, 40, 54], have emerged as leading approaches. Additionally, representation-based strategies [30, 33], optimization-oriented methods [22, 27, 50], and various hybrid or novel methods [19, 29, 52] are progressively expanding the scope of incremental object detection research. Nevertheless, compared to the extensive body of work on incremental classification, incremental learning for object detection remains relatively understudied [46].

2.2. Prompt-based Incremental Learning

Visual prompts are a parameter-efficient fine-tuning technique to adapt pre-trained models to new data [20]. The initial method L2P [49] demonstrated the feasibility of applying visual prompts to IL. They learned a pool of these visual prompts and a corresponding key for each prompt. The visual prompts are selected using cosine similarity between the classification token and this key. Following methods improve on this by distinguishing between general and expert prompts [48], employing non-shared prompt pools [47], prompt-selection through k-nearest neightbor search [47], attention-based prompt combination [41], separate learning objectives [45], or generating prompts using meta-networks [23, 31, 53]. A further overview is provided by Wang et al. [46] and Zhou et al. [60]. However, these prompt-based methods are not evaluated on other IL computer vision types than classification.

3. Preliminary

3.1. Domain Incremental Object Detection

We study the problem of domain incremental object detection, where a model is exposed to a sequence of tasks, *i.e*. domains [46, 47, 58]. At step t, the model learns task $\mathcal{T}_t$ using the dataset $\mathcal{D}_t = (\mathcal{X}_t, \mathcal{Y}_t)$, where the image set $\mathcal{X}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$ consists of n_t images. Each image $\mathbf{x}_i^t$ has dimensions $\mathbf{x}_i^t \in \mathbb{R}^{H_i^t \times W_i^t \times C_i^t}$, with H_i^t, W_i^t, and C_i^t denoting height, width, and channel count, respectively.

The annotation set $\mathcal{Y}_t = \{\mathbf{y}_i^t\}_{i=1}^{n_t}$ corresponds to these images, where each $\mathbf{y}_i^t$ is a collection of object instances: $\mathbf{y}_i^t = \{(c_{i,j}^t, \mathbf{b}_{i,j}^t)\}_{j=1}^{m_i^t}$. Here, $c_{i,j}^t \in \mathcal{C}$ represents the class label of the j-th object in image $\mathbf{x}_i^t$, with $\mathcal{C}$ being the category set, which is fixed for this domain IOD setting, and $\mathbf{b}_{i,j}^t \in \mathbb{R}^4$ denotes the bounding box coordinates. The number of annotated objects m_i^t may vary across images.

During training, the model has access to the task identity, but this information is not provided at test time. A model trained on task $\mathcal{T}_t$ using the data $\mathcal{D}_t$ is denoted by $\mathcal{M}_t$.

3.2. Visual Prompt Tuning

A visual prompt is a set of learnable parameters $\mathbf{p} \in \mathbb{R}^{L_p \times D}$, where L_p is the prompt length, *i.e*. the number of prompts, and D is the embedding dimension [20, 48]. The backbone itself is kept frozen, and the visual prompts are incorporated into it and optimized during training. There are two prominent ways to incorporate the visual prompts into the backbone.

- **Prompt Tuning (Pro-T).** The prompts are prepended to the key h_K, query h_Q and value h_V of the multi-head self-attention (MSA) layer.

$$f_{\text{prompt}}^{\text{Pro-T}} = \text{MSA}([p; h_Q], [p; h_K], [p; h_V]) \quad (1)$$

  Here, $[\cdot;\cdot]$ is the concatenation operation along the sequence length. The output sequence, compared to the non-prompted MSA, is extended by the length of the prompt.
- **Prefix Tuning (Pre-T).** The prompt is split into two parts that are prepended to the key and value, *i.e*. $p_K, p_V \in \mathbb{R}^{L_p/2 \times D}$.

$$f_{\text{prompt}}^{\text{Pre-T}} = \text{MSA}(h_Q, [p_K; h_K], [p_V; h_V]) \quad (2)$$

  The length of the output sequence remains unchanged by the visual prompts.

More details can be found here [20, 48].

4. Experiments

4.1. Setup

Model. We use the EVA-02 vision transformer [12] in its *big* configuration. We include the prompts in the positional embedding but exclude them from the rotary embedding. We repeat the prompts on the window partitioning layers by the number of windows. We use the COCO pre-trained weights and freeze the backbone, region proposal network, and head, leaving only the output layer trainable.

Optimization. We employ the AdamW optimizer with a learning rate of 0.001 and cosine learning rate decay. We train each task for 1,000 iterations and a batch size of 10.

4.1.1. Methods

We select three prominent methods for the evaluation: L2P [49], DualPrompt [48], and S-prompt [47]. While these are not the state-of-the-art (SOTA), their simplicity allows

Table 1. Results for different prompting techniques and prompt-based IL methods on the D-RICO benchmark. Joint and individual training represent the upper bounds, Naïve FT the lower bound, and the two replay configurations are strong baselines. The three prompt-based IL methods fall behind even 1% replay. Best IL approach in bold.

			Domain RICO			
Method	**Prompt style**	**Freeze Head after 1. Task**	$\overline{\text{mAP}}$ ↑	FM ↓	FWT ↑	IM ↑
Joint Training	No Prompt	✗	25.45	-	-	-
	Shallow Prompt	✗	26.39	-	-	-
	Deep Prompt	✗	29.55	-	-	-
Individual Training	No Prompt	✗	26.92	-	-	-
	Shallow Prompt	✗	28.98	-	-	-
	Deep Prompt	✗	33.12	-	-	-
Naïve FT	No Prompt	✗	16.20	13.17	-7.31	-2.81
	Shallow Prompt	✗	20.88	10.38	-4.25	-0.36
	Deep Prompt	✗	21.98	16.60	2.54	5.71
	No Prompt	✓	23.49	0	-11.32	-6.05
	Shallow Prompt	✓	23.23	2.54	-9.19	-4.27
	Deep Prompt	✓	22.89	14.53	1.66	4.96
Replay 1%	No Prompt	✗	21.44	7.08	-6.81	-2.44
	Shallow Prompt	✗	23.16	6.79	-4.96	-0.95
	Deep Prompt	✗	26.55	10.74	2.43	5.60
	Shallow Prompt	✓	23.27	2.30	-9.29	-4.42
	Deep Prompt	✓	26.94	9.71	1.77	5.16
Replay 10%	No Prompt	✗	25.41	2.81	-6.11	-1.88
	Shallow Prompt	✗	26.79	3.64	-3.64	0.14
	Deep Prompt	✗	**31.62**	4.63	**2.60**	**5.76**
	Shallow Prompt	✓	24.41	**0.89**	-9.29	-4.41
	Deep Prompt	✓	31.15	3.59	1.11	**5.76**
L2P [49]		✗	20.92	10.33	-4.28	-0.35
		✓	23.28	1.89	-9.80	-4.76
DualPrompt [48]		✗	18.61	12.29	-5.16	-1.11
		✓	23.81	1.07	-9.94	-4.91
S-Prompt [47]		✗	20.71	10.27	-4.61	-0.62
		✓	22.78	1.36	-10.86	-5.66

for a clearer understanding of the problem and provides valuable insights. We consider two different configurations: fixing the head after the first task and continuing to train the head.

We compare these prompt-based IL methods to a wide variety of reference baselines [36]:

- **Joint Training** merges all tasks into a single training dataset and trains a single model on these. The test datasets are separate.
- **Individual Training** trains and tests a separate model for each task.
- **Naïve finetuning** (FT) trains a single model sequentially on the tasks without any IL method.
- **Replay** keeps a portion (1% and 10% in this case) for the sequential tasks to train on new and some old data at the same time.

We consider different configurations for these reference baselines:

- **Freeze Head after 1. Task** to reduce model plasticity.
- **No Prompt** uses the standard EVA-02 model without modifications.
- **Shallow Prompt** uses a trainable 50 prompts and prepends them to the image embeddings before the first attention block [20].
- **Deep Prompt** learns 100 prompts for each layer and prepends them to the image embeddings [20].

All settings employ prompt tuning, with prefix tuning being used only in DualPrompt.

4.2. Benchmark

We employ the D-RICO benchmark [36] as it offers the most diverse domain distribution shifts. It consists of 14 different datasets from which 15 tasks are created. These datasets encompass various camera sensors (RGB, thermal, gated, and event), lenses, viewpoints, time of day, weather conditions, and both real and synthetic domains. The output distribution also varies in terms of bounding box location, aspect ratio, and class ratios. Additionally, due to the origin of multiple datasets, the label quality and policy vary. Leading, all together, to the most diverse domain IOD benchmark, providing a complex challenge for any method.

Table 2 lists the tasks, their names, classes, and brief descriptions.

4.3. Evaluation Metrics

To assess IL performance, we adopt widely used metrics [5, 36, 46], using mean Average Precision (mAP) as the primary evaluation criterion [32]. Our evaluation focuses on three aspects:

1. **Overall effectiveness.** We measure aggregate performance with the average mAP, denoted as $\overline{\text{mAP}}$. Let $\text{mAP}_{k,j}$ represent the mAP achieved on test set $\mathcal{D}_j$ of task $\mathcal{T}_j$ after completing training on task $\mathcal{T}_k$ (where $j \leq k$). The cumulative performance after task k is defined as:

$$\overline{\text{mAP}}_k = \frac{1}{k} \sum_{j=1}^{k} \text{mAP}_{k,j}, \tag{3}$$

   where larger values indicate better retention and generalization across tasks.

2. **Retention and forgetting.** We evaluate memory stability via the *forgetting measure* (FM), which captures the decline in a model's performance on earlier tasks. After training on task k, the forgetting metric is computed as:

$$\text{FM}_k = \frac{1}{k-1} \sum_{j=1}^{k-1} \max_{1 \leq l \leq k-1} \left(\text{mAP}_{l,j} - \text{mAP}_{k,j}\right). \tag{4}$$

   A higher FM value reflects increased forgetting, while negative values suggest performance gains on prior tasks.

3. **Adaptability and transfer.** A model's ability to learn new tasks effectively is characterized by two complementary metrics:
   (a) *Forward transfer* (FWT) quantifies how previously acquired knowledge benefits learning a new task. It is calculated as:

$$\text{FWT}_k = \frac{1}{k-1} \sum_{j=2}^{k} \left(\text{mAP}_{j,j} - \text{mAP}'_j\right), \tag{5}$$

   where mAP'_j denotes the performance of an independently trained model on task $\mathcal{T}_j$. Positive FWT indicates improved learning due to prior experience.
   (b) *Intransigence* (IM) assesses the difficulty in learning new tasks relative to a jointly trained model. It is defined as:

$$\text{IM}_k = \frac{1}{k} \sum_{j=1}^{k} \left(\text{mAP}_{j,j} - \text{mAP}^*_j\right), \tag{6}$$

   where mAP^*_j corresponds to the mAP obtained from a model trained on all task data $\cup_{j=1}^{T} \mathcal{D}_j$ simultaneously. A positive IM implies greater adaptability than joint training.

At the conclusion of all T tasks, we denote the final metric values as $\overline{\text{mAP}} = \overline{\text{mAP}}_T$, $\text{FM} = \text{FM}_T$, $\text{FWT} = \text{FWT}_T$, and $\text{IM} = \text{IM}_T$.

The overarching objective is for IL models to surpass both standalone and joint models by leveraging inter-task transfer, ideally satisfying $\overline{\text{mAP}} > \frac{1}{T} \sum_{j=1}^{T} \text{mAP}'_j$, which necessitates high adaptability and minimal forgetting.

4.4. Results

We first present the main results of three prompt-based methods and reference baselines on the D-RICO benchmark, followed by additional analyses on initialization and prompt length .We choose a diverse subset of five tasks, *i.e.* $[1, 2, 3, 11, 15]$, from the 15 D-RICO tasks for the main results and all 15 tasks in the subsequent experiments.

4.4.1. Main Results

The main results on the D-RICO benchmark are shown in Table 1 and Figure 1. Among the three prompt-based IL methods, L2P achieves the highest performance when the output layer is not frozen, while DualPrompt slightly outperforms the other two methods when the output layer is fixed after the first task. Regarding forgetting, DualPrompt is also the lowest.

The three prompt-based IL methods perform similarly to Naïve FT and lag substantially behind replay at both 1% and 10%. In Figure 1, this becomes more obvious where they show high forgetting while having mediocre overall performance and plasticity. However, as all three methods do not employ deep prompting, in a fair comparison to shallow prompting, they achieve a similar performance to replay 1%, though Naïve FT is also close to that.

Fixing the output layer generally benefits all IL settings except for the 10% replay scenario. Specifically, weaker methods such as Naïve FT, replay 1%, L2P, DualPrompt, and S-Prompt all benefit from reduced model plasticity, as their counterparts with non-fixed output layers exhibit lower performance in terms of $\overline{\textbf{mAP}}$ and FM. However, strong regularization via 10% replay benefits from increased plasticity, enabling it to surpass individually trained models in both shallow and deep prompt scenarios.

Overall, deep prompting consistently outperforms shallow prompting regarding $\overline{\textbf{mAP}}$ and FWT, although shallow prompting demonstrates lower FM. The two plasticity metrics (FWT and IM) show an increase in model adaptability. For Naïve FT, deep and shallow prompting yield similar $\overline{\textbf{mAP}}$, highlighting a trade-off between stability (FM) and plasticity (FWT). Employing prompts generally outperforms the no-prompt condition. However, when the output layer is fixed (i.e., no further learning occurs after the initial task), the $\overline{\textbf{mAP}}$ performance of the no-prompt condition becomes similar to the three prompt-based methods. This further illustrates that these standard methods are not sufficiently competitive on this challenging benchmark.

Table 2. Description for D-RICO benchmark that consists of 15 tasks from 14 different datasets *incorporating* variations in multiple different aspects.

Task Number	Task Name	Dataset	Classes	Short Description
1	*daytime*	nuImages [2]	person, bicycle, vehicle	urban, daylight, real-world, vehicle-mounted, Singapore
2	*thermal*	Teledyne FLIR [13]	person, bicycle, vehicle	thermal, urban, varying lighting, weather conditions
3	*fisheye fix*	FishEye8K [17]	person, vehicle	fisheye, daytime, urban traffic, Taiwan, wide-angle, multi-camera
4	*drone*	VisDrone [61]	person, bicycle, vehicle	drone, urban and rural, variable density, different lighting, 14 cities
5	*simulation*	SHIFT [43]	person, bicycle, vehicle	synthetic, urban driving, CARLA, daytime, clear weather
6	*fisheye car*	WoodScape [56]	person, vehicle	fisheye, vehicle-mounted, driving perspectives, multiple positions
7	*RGB + thermal fusion*	SMOD [6]	person, bicycle, vehicle	RGB-thermal fusion using IFCNN [59]
8	*video game*	Sim10k [21]	vehicle	synthetic, urban, GTA V, diverse driving scenarios
9	*nighttime*	BDD100K [57]	person, bicycle, vehicle	urban, nighttime, perception challenge, street lighting
10	*fisheye indoor*	LOAF [55]	person	fisheye, indoor, overhead, 360° view, surveillance
11	*gated*	DENSE [1]	person, vehicle	gated, urban, various conditions, depth-enhanced imaging
12	*photoreal. simulation*	Synscapes [51]	person, vehicle	photorealistic, synthetic, urban, physically based rendering
13	*thermal fisheye indoor*	TIMo [39]	person	thermal fisheye, indoor, human actions, multiple perspectives
14	*inclement*	DENSE [1]	person, vehicle	fog, snow, rain, adverse weather
15	*event camera*	DSEC [15, 16]	person, bicycle, vehicle	event-based, driving, varied lighting, RGB overlay

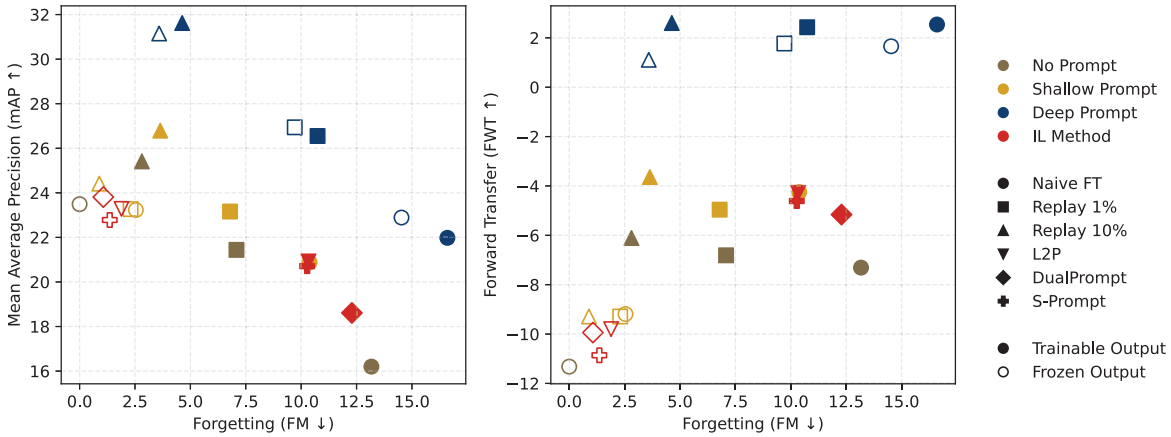

Figure 1. Incremental learning results on D-RICO benchmark. The left figure shows overall performance $\overline{\mathbf{mAP}}$ versus the forgetting (FM) and the right shows plasticity (FWT) versus FM. The three prompt-based IL methods are far from the optimal of high plasticity and low forgetting (upper left corner).

4.4.2. Prompt Length

It is expected that different tasks require varying prompt lengths depending on their diversity. To illustrate this, we train each task in the D-RICO benchmark with different prompt lengths (1, 5, 10, 25, 100, 250, 500) to identify the optimal length for each. The results shown in Table 3 confirms this across three different prompting techniques. It is evident that some tasks perform well with a single prompt, while others require up to 500. Choosing the best prompt length for each task slightly increases the final mAP. However, the difference compared to the next-best fixed prompt length is minimal.

Figure 2 shows a histogram of how often a prompt length yields the best outcome. When there's a tie, the shorter length is selected because it's more computationally efficient and thus preferred. It is clear that the optimal prompt length depends on the prompting style (shallow versus deep and remove versus keep prompt) and the task. Generally, deep prompting can better utilize longer prompt lengths compared to shallow prompting. If the prompt is removed after the first layer, i.e., it only influences the first MSA, longer prompt lengths work better than when the prompt is kept in the model.

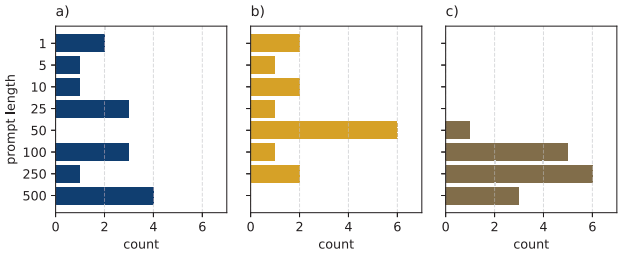

Figure 2. Count for how often a prompt length led to the best mAP in each of the three prompting categories. Plot a) shows result for shallow prompting where the prompt is removed after the first layer, b) shallow with keeping the prompt and c) deep prompt. The results demonstrate that larger prompt length work well for deep prompting, and shallow prompting requires a bit less.

Table 3. Analysis of optimal prompt length across three prompt techniques for each D-RICO task shows that not all prompt configurations outperform training without prompts. Generally, shallow prompts offer only marginal gains. Selecting the optimal prompt length per task yields the best average performance, though the improvement over a fixed prompt length is minimal.

		Shallow Prompt (Remove Prompt)									Shallow Prompt (Keep Prompt)									Deep Prompt								
Task	0	1	5	10	25	50	100	250	500	Best	1	5	10	25	50	100	250	500	Best	1	5	10	25	50	100	250	500	Best
1	41.2	41.0	41.3	41.2	**41.6**	41.2	41.3	41.2	41.4	41.6	41.2	**41.4**	41.2	41.2	41.1	41.0	41.0	40.5	41.4	42.5	43.4	43.6	43.9	44.4	43.9	**44.5**	44.5	44.5
2	33.2	33.2	34.1	34.1	34.2	34.6	34.5	34.8	**35.3**	35.3	33.4	33.4	34.2	34.8	34.7	35.0	**35.3**	34.6	35.3	37.7	38.5	39.1	39.3	39.9	**40.3**	39.8	39.9	40.3
3	20.3	19.6	19.7	20.0	20.0	20.5	20.5	20.3	**20.6**	20.6	20.0	20.1	20.3	20.3	20.4	20.3	**20.7**	20.6	20.7	23.1	24.0	24.2	24.8	**25.0**	24.8	25.0	24.4	25.0
4	18.7	18.7	18.6	18.6	18.7	18.6	18.6	**18.8**	18.7	18.8	18.7	18.8	18.8	18.9	18.9	**19.0**	18.8	18.8	19.0	20.2	20.4	20.8	20.9	21.1	**21.2**	21.1	21.2	21.2
5	30.7	**30.2**	29.8	29.7	29.9	29.6	29.6	29.1	29.7	30.2	30.1	30.1	**30.3**	30.0	29.8	29.7	29.6	29.3	30.3	31.1	31.6	32.0	32.3	32.5	32.6	**32.7**	32.5	32.7
6	39.9	39.8	39.7	39.7	**40.0**	39.8	39.7	39.9	39.9	40.0	39.8	39.9	39.9	39.9	**40.1**	40.0	39.9	39.3	40.1	41.1	41.7	41.9	42.3	42.0	42.3	42.2	**42.4**	42.4
7	44.9	44.6	44.6	**45.0**	44.9	45.0	44.9	45.0	44.7	45.0	44.9	44.8	**45.0**	44.8	45.0	44.2	44.2	43.4	45.0	47.1	49.8	50.0	50.4	50.4	50.4	**50.8**	50.3	50.8
8	44.8	44.9	45.0	45.0	45.0	44.8	**45.1**	44.7	44.7	45.1	**45.2**	45.0	45.1	45.2	45.1	44.8	44.6	44.4	45.2	45.8	46.1	46.5	46.6	46.3	46.5	46.1	**46.8**	46.8
9	20.5	21.0	21.2	20.7	21.5	20.8	**21.8**	21.5	21.7	21.8	20.4	21.3	21.1	21.3	**21.5**	21.4	20.9	21.1	21.5	21.3	22.2	22.3	22.9	22.7	23.1	**23.4**	22.7	23.4
10	37.3	37.7	37.9	37.4	**38.2**	37.6	37.5	38.0	37.4	38.2	37.4	37.9	37.9	37.8	**38.0**	37.2	36.9	36.9	38.0	43.1	43.9	43.4	43.2	44.3	**44.4**	44.4	43.4	44.4
11	27.7	27.8	**28.2**	27.5	27.8	27.8	27.4	27.7	27.5	28.2	**28.1**	27.2	27.3	26.6	27.0	24.7	27.4	21.0	28.1	30.7	31.5	32.2	32.5	32.9	**33.5**	33.2	33.2	33.5
12	24.6	**24.1**	24.0	23.8	24.0	24.1	24.0	23.9	24.0	24.1	24.1	24.2	24.1	24.3	**24.5**	24.5	24.4	24.3	24.5	25.4	25.9	26.1	26.2	26.4	26.5	26.6	**26.7**	26.7
13	69.6	72.1	71.4	70.9	72.4	72.3	72.6	72.0	**73.6**	73.6	69.8	70.5	72.6	72.9	**73.0**	73.0	71.1	71.8	73.0	80.2	82.6	82.7	83.4	83.6	**84.7**	84.3	84.0	84.7
14	48.5	48.6	48.6	48.5	48.6	48.5	48.6	48.5	**48.7**	48.7	48.7	48.6	48.6	**48.7**	48.6	48.3	48.4	47.9	48.7	49.4	50.0	50.0	51.0	50.9	50.9	**51.1**	51.1	51.1
15	12.2	17.1	20.1	18.4	19.2	19.3	**21.3**	18.9	20.4	21.3	15.8	18.8	18.3	21.0	**21.7**	18.7	20.0	20.1	21.7	20.3	23.5	21.3	23.4	23.2	23.1	**25.4**	23.2	25.4
Mean	34.3	34.7	34.9	34.7	35.1	35.0	35.2	35.0	35.2	**35.5**	34.5	34.8	35.0	35.2	35.3	34.8	34.9	34.3	**35.5**	37.3	38.3	38.4	38.9	39.0	39.1	39.4	39.1	**39.5**

4.4.3. Prompt Initialization

Previous works on visual prompt methods for IL used uniform prompt initialization with random values between -1 and 1. We noticed in preliminary experiments that we can achieve better results with smaller intervals. To study this further, we run experiments for the initializations values $\text{init} \in \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, meaning prompt initialization with uniform random values in the interval $[-\text{init}, \text{init}]$, for different configurations. For the different configurations we use prompt lengths $L_P \in \{1, 5, 10, 25, 50, 100, 250, 500\}$ and injection layers $\text{inject} \in \{[0], [0, 1, 2, 3], [0, 1, 2, 3, 4, 5, 6], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12], [7, 8, 9, 10, 11, 12]\}$. We only train on task 4.

Figure 3 displays the average results for various initialization configurations. It is evident that the commonly used interval $[-1, 1]$ does not produce the best outcomes. Below 10^{-2}, results level off, indicating that for low uniform initialization values, the specific value chosen has little effect on the outcome. The standard deviation remains similar across all low initialization values and decreases slightly towards 1.

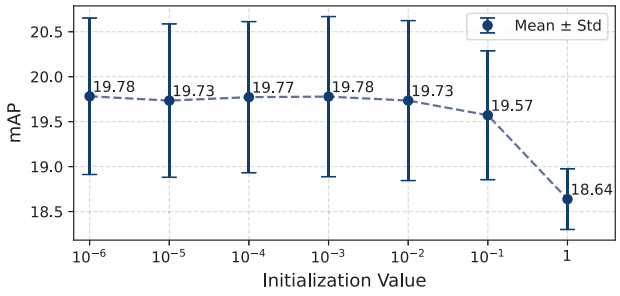

Figure 3. Results of different prompt initialization intervals $[-\text{init}, \text{init}]$ averaged over various prompt lengths and injection layers for task 4 from D-RICO. For lower values, the results stabilize and are better than for larger intervals.

5. Discussion

In this section, we collectively summarize and discuss these findings, with key takeaways provided in the text box.

The results of DualPrompt in Table 1 demonstrate the general feasibility of employing prompt-based IL methods for domain IOD. However, all three tested methods underperform compared to randomly replaying data, highlighting the necessity for more advanced prompt-based methods. A wide variety of methods developed for classification could be explored in future work.

The study of the reference baselines indicates that deep prompting significantly outperforms shallow prompting. Thus, future research should focus on deep prompts to enhance overall performance and increase plasticity.

As observed, replaying just 1% of data from previous tasks represents a simple yet robust baseline. Expanding the replay buffer to 10% further reduces forgetting and improves overall performance.

Naive FT, replay 1%, L2P, DualPrompt, and S-Prompt all benefit from fixing the output layer after the initial task. Strong regularization, as employed in the 10% replay scenario, further improves performance due to increased available plasticity. Therefore, future methods should also consider adaptations at the output layer, as modifications solely in the feature space are insufficient for achieving optimal IL performance.

Determining the optimal prompt length is not straightforward, as it varies depending on the specific prompting technique and task. While selecting an individual length per task provides minor advantages, the benefits currently do not justify the complexity and additional hyperparameter tuning required. However, this aspect could become relevant in future benchmarks or practical applications. Generally, object detection requires longer prompts compared to classification tasks.

In contrast, the influence of prompt initialization on performance is significant. Results presented in Figure 3 suggest that initializing with smaller random values from the uniform interval $[-10^{-2}, 10^{-2}]$ yields superior and more stable performance. This differs from prompt-based classification methods, where random initialization typically occurs within a larger interval, such as $[-1, 1]$. A better initialization scheme can notably improve results.

Future investigations and experiments should assess the performance of class incremental learning (CIL) and few-shot IL for DIL and CIL. CIL is particularly challenging, as prompt-based methods only modify the backbone, necessitating an additional mechanism to address forgetting in an expanding head. Since the studied D-RICO benchmark surpasses existing benchmarks in diversity, we anticipate that the results will apply to these less diverse benchmarks.

Key Takeaways

Feasibility. Visual-prompt methods studied here provide minimal help in mitigating catastrophic forgetting for domain incremental learning.

Best Method. DualPrompt [48] performs best among the tested prompt-based approaches.

Deep vs. Shallow. Deep prompts significantly outperform shallow prompts.

Output Layer. Unfreezing the output layer, paired with strong regularization, enhances performance by increasing plasticity. Weak methods profit from freezing the output layer after the first task.

Replay. Replaying even 1% of previous data surpasses prompt-based IL methods, especially in combination with deep prompting.

Prompt Length. Deep prompting benefits more from longer prompts. Using an individual prompt length for all tasks yields only minimal improvement in the results.

Prompt Initialization. Random initialization values for the prompts within $[-10^{-2}, 10^{-2}]$ or narrower yield optimal performance.

6. Conclusion

This work presents the first comprehensive analysis of prompt-based IL methods for object detection. We evaluated three classification-derived approaches—L2P [11], DualPrompt [48], and S-Prompt [47]—against a range of strong reference baselines on the challenging D-RICO benchmark [36]. Our findings confirm the general feasibility of applying prompt-based IL to object detection, with DualPrompt achieving the highest performance among the prompt-based methods. However, all evaluated methods are still outperformed by simple replay-based strategies, underscoring the need for further innovation in prompt design and learning mechanisms. We believe that our empirical insights will serve as valuable guidance for advancing prompt-based IL methods in object detection.

References

[1] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather. In *CVPR*, pages 11679–11689, 2020. 5

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2020. 5

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *PAMI*, 43(5):1483–1498, 2021. 2

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End Object Detection with Transformers. In *ECCV*, pages 213–229. 2020. 2

[5] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *ECCV*, pages 556–572. 2018. 4

[6] Zizhao Chen, Yeqiang Qian, Xiaoxiao Yang, Chunxiang Wang, and Ming Yang. AMFD: Distillation via Adaptive Multimodal Fusion for Multispectral Pedestrian Detection. *arXiv preprint arXiv:2405.12944*, 2024. 5

[7] Na Dong, Yongqiang Zhang, Mingli Ding, and Yancheng Bai. Class-incremental object detection. *Pattern Recognition*, 139:109488, 2023. 2

[8] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Incremental-DETR: Incremental Few-Shot Object Detection via Self-Supervised Learning. *AAAI*, 37(1):543–551, 2023. 2

[9] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Towards Non Co-occurrence Incremental Object Detection with Unlabeled In-the-Wild Data. *Int J Comput Vis*, 2024. 2

[10] Lixuan Du, Rongyu Zhang, and Xiaotian Wang. Overview of two-stage object detection algorithms. *J. Phys.: Conf. Ser.*, 1544(1):012033, 2020. 2

[11] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to Prompt for Open-Vocabulary Object Detection with Vision-Language Model, 2022. 7

[12] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A Visual Representa-

tion for Neon Genesis. *arXiv preprint arXiv:2303.11331*, (arXiv:2303.11331), 2023. 2

[13] Teledyne FLIR. FREE Teledyne FLIR Thermal Dataset for Algorithm Training. https://www.flir.com/oem/adas/adas-dataset-form/, Accessed: 01.12.2024. 5

[14] Sumyung Gang, Daewon Chung, and Joonjae Lee. Predictive Distillation Method of Anchor-Free Object Detection Model for Continual Learning. *Applied Sciences*, 12(13): 6419, 2022. 2

[15] Daniel Gehrig and Davide Scaramuzza. Low-latency automotive vision with event cameras. *Nature*, 629(8014):1034–1040, 2024. 5

[16] Mathias Gehrig, Willem Aarents, Daniel Gehrig, and Davide Scaramuzza. DSEC: A Stereo Event Camera Dataset for Driving Scenarios. *IEEE Robot. Autom. Lett.*, 6(3):4947–4954, 2021. 5

[17] Munkhjargal Gochoo, Munkh-Erdene Otgonbold, Erkhembayar Ganbold, Jun-Wei Hsieh, Ming-Ching Chang, Ping-Yang Chen, Byambaa Dorj, Hamad Al Jassmi, Ganzorig Batnasan, Fady Alnajjar, Mohammed Abduljabbar, and Fang-Pang Lin. FishEye8K: A Benchmark and Dataset for Fisheye Camera Object Detection. In *CVPR Workshops*, pages 5305–5313, 2023. 5

[18] Muhammad Hussain. YOLOv1 to v8: Unveiling Each Variant–A Comprehensive Review of YOLO. *IEEE Access*, 12:42816–42833, 2024. 2

[19] M. Tahasanul Ibrahim, Nikhil Limaye, and Andreas Schwung. Node Reservation Based Incremental Learning Network for Object Detection. In *ICIT*, pages 1–6, 2024. 2

[20] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning. In *ECCV*, pages 709–727. 2022. 2, 3

[21] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks? *arXiv preprint arXiv:1610.01983*, 2017. 5

[22] K. J. Joseph, Jathushan Rajasegaran, Salman Khan, Fahad Shahbaz Khan, and Vineeth N. Balasubramanian. Incremental Object Detection via Meta-Learning. *PAMI*, 44 (12):9209–9216, 2022. 2

[23] Dahuin Jung, Dongyoon Han, Jihwan Bang, and Hwanjun Song. Generating Instance-level Prompts for Rehearsal-free Continual Learning. In *ICCV*, pages 11813–11823, 2023. 2

[24] Mengxue Kang, Jinpeng Zhang, Jinming Zhang, Xiashuang Wang, Yang Chen, Zhe Ma, and Xuhui Huang. Alleviating Catastrophic Forgetting of Incremental Object Detection via Within-Class and Between-Class Knowledge Distillation. In *ICCV*, pages 18894–18904, 2023. 2

[25] Dongwan Kim and Bohyung Han. On the Stability-Plasticity Dilemma of Class-Incremental Learning. In *CVPR*, pages 20196–20204, 2023. 1

[26] 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 *CVPR*, pages 28772–28781, 2024. 2

[27] Haifeng Li, Ye Chen, Zhenshi Zhang, and Jian Peng. RAISE: Rank-Aware Incremental Learning for Remote Sensing Object Detection. *Symmetry*, 14(5):1020, 2022. 2

[28] Yuyang Liu, Yang Cong, Dipam Goswami, Xialei Liu, and Joost Van De Weijer. Augmented Box Replay: Overcoming Foreground Shift for Incremental Object Detection. In *ICCV*, pages 11333–11343, 2023. 2

[29] Yaoyao Liu, Bernt Schiele, Andrea Vedaldi, and Christian Rupprecht. Continual Detection Transformer for Incremental Object Detection. In *CVPR*, pages 23799–23808, 2023. 2

[30] Xiaonan Lu, Wenhui Diao, Junxi Li, Yidan Zhang, Peijin Wang, Xian Sun, and Kun Fu. Few-Shot Incremental Object Detection in Aerial Imagery via Dual-Frequency Prompt. *IEEE Trans. Geosci. Remote Sensing*, 62:1–17, 2024. 2

[31] Yue Lu, Shizhou Zhang, De Cheng, Guoqiang Liang, Yinghui Xing, Nannan Wang, and Yanning Zhang. Training Consistent Mixture-of-Experts-Based Prompt Generator for Continual Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):19152–19160, 2025. 2

[32] Angelo G. Menezes, Gustavo de Moura, Cézanne Alves, and André C. P. L. F. de Carvalho. Continual Object Detection: A review of definitions, strategies, and challenges. *arXiv preprint arXiv:2205.15445*, 2022. 1, 2, 4

[33] Jianwen Mo, Ronghua Zou, and Hua Yuan. Multi-Level Foreground Prompt for Incremental Object Detection. *IEEE Access*, 13:4048–4066, 2025. 2

[34] 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 *ECCV*, pages 463–480. 2025. 2

[35] Riccardo De Monte, Davide Dalle Pezze, Marina Ceccon, Francesco Pasti, Francesco Paissan, Elisabetta Farella, Gian Antonio Susto, and Nicola Bellotto. Replay Consolidation with Label Propagation for Continual Object Detection. *arXiv preprint arXiv:2409.05650*, 2024. 2

[36] Matthias Neuwirth-Trapp, Maarten Bieshaar, Danda Pani Paudel, and Luc Van Gool. Rico: Two realistic benchmarks and an in-depth analysis for incremental learning in object detection, 2025. 1, 3, 4, 7

[37] Peisheng Qian, Kai Zheng, Cen Chen, Zhongyao Cheng, Li Wang, and Hui Li Tan. Contrastive R-CNN for Incremental Learning in Object Detection. *SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta*, pages 557–563, 2022. 2

[38] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *PAMI*, 39(6):1137–1149, 2017. 2

[39] Pascal Schneider, Yuriy Anisimov, Raisul Islam, Bruno Mirbach, Jason Rambach, Didier Stricker, and Frédéric Grandidier. TIMo—A Dataset for Indoor Building Monitoring with a Time-of-Flight Camera. *Sensors*, 22(11):3992, 2022. 5

[40] Jeng-Lun Shieh, Qazi Mazhar Ul Haq, Muhamad Amirul Haq, Said Karam, Peter Chondro, De-Qin Gao, and Shanq-Jang Ruan. Continual Learning Strategy in One-Stage Object Detection Framework Based on Experience Replay for

Autonomous Driving Vehicle. *Sensors*, 20(23):6777, 2020. 2

[41] 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 *CVPR*, pages 11909–11919, 2023. 2

[42] Xiang Song, Yuhang He, Songlin Dong, and Yihong Gong. Non-exemplar Domain Incremental Object Detection via Learning Domain Bias. *AAAI*, 38(13):15056–15065, 2024. 2

[43] Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. SHIFT: A Synthetic Driving Dataset for Continuous Multi-Task Domain Adaptation. In *CVPR*, pages 21339–21350, 2022. 5

[44] Baptiste Wagner, Denis Pellerin, and Sylvain Huet. Forgetting Analysis by Module Probing for Online Object Detection with Faster R-CNN. In *EUSIPCO*, pages 576–580, 2024. 2

[45] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical Decomposition of Prompt-Based Continual Learning: Rethinking Obscured Sub-optimality. 2024. 2

[46] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A Comprehensive Survey of Continual Learning: Theory, Method and Application. *PAMI*, pages 1–20, 2024. 1, 2, 4

[47] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-Prompts Learning with Pre-trained Transformers: An Occam's Razor for Domain Incremental Learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022. 1, 2, 3, 7

[48] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary Prompting for Rehearsal-Free Continual Learning. In *ECCV*, pages 631–648. 2022. 1, 2, 3, 7

[49] 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 *CVPR*, pages 139–149, 2022. 1, 2, 3

[50] Zezhou Wang, Guitao Cao, Xidong Xi, and Jiangtao Wang. OpenNet: Incremental Learning for Autonomous Driving Object Detection with Balanced Loss. *SMC*, pages 2675–2682, 2023. 2

[51] Magnus Wrenninge and Jonas Unger. Synscapes: A Photorealistic Synthetic Dataset for Street Scene Parsing. *arXiv preprint arXiv:1810.08705*, 2018. 5

[52] Binbin Yang, Xinchi Deng, Han Shi, Changlin Li, Gengwei Zhang, Hang Xu, Shen Zhao, Liang Lin, and Xiaodan Liang. Continual Object Detection via Prototypical Task Correlation Guided Gating Mechanism. In *CVPR*, pages 9245–9254, 2022. 2

[53] Chengyi Yang, Wentao Liu, Shisong Chen, Jiayin Qi, and Aimin Zhou. Generating Prompts in Latent Space for Rehearsal-free Continual Learning. In *ACM Multimedia*, pages 8913–8922, 2024. 2

[54] Dongbao Yang, Yu Zhou, Xiaopeng Hong, Aoting Zhang, and Weiping Wang. One-Shot Replay: Boosting Incremental Object Detection via Retrospecting One Object. *AAAI*, 37 (3):3127–3135, 2023. 2

[55] Lu Yang, Liulei Li, Xueshi Xin, Yifan Sun, Qing Song, and Wenguan Wang. Large-Scale Person Detection and Localization using Overhead Fisheye Cameras. *arXiv preprint arXiv:2307.08252*, 2023. 5

[56] Senthil Yogamani, Ciaran Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O'Dea, Michal Uricar, Stefan Milz, Martin Simon, Karl Amende, Christian Witt, Hazem Rashed, Sumanth Chennupati, Sanjaya Nayak, Saquib Mansoor, Xavier Perroton, and Patrick Perez. WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving. *arXiv preprint arXiv:1905.01489*, 2021. 5

[57] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. *arxiv preprint arXiv:1805.04687*, 2020. 5

[58] Jinghua Zhang, Li Liu, Olli Silvén, Matti Pietikäinen, and Dewen Hu. Few-Shot Class-Incremental Learning for Classification and Object Detection: A Survey. *PAMI*, 2025. 2

[59] Yu Zhang, Yu Liu, Peng Sun, Han Yan, Xiaolin Zhao, and Li Zhang. IFCNN: A general image fusion framework based on convolutional neural network. *Information Fusion*, 54: 99–118, 2020. 5

[60] 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*, (arXiv:2401.16386), 2024. 1, 2

[61] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and Tracking Meet Drones Challenge. *PAMI*, 44(11):7380–7399, 2022. 5