PseDet: Revisiting the Power of Pseudo Label in Incremental Object Detection

Qiuchen Wang[1], Zehui Chen[1], Chenhongyi Yang[2], Jiaming Liu[3], Zhenyu Li[4], Feng Zhao[1]*

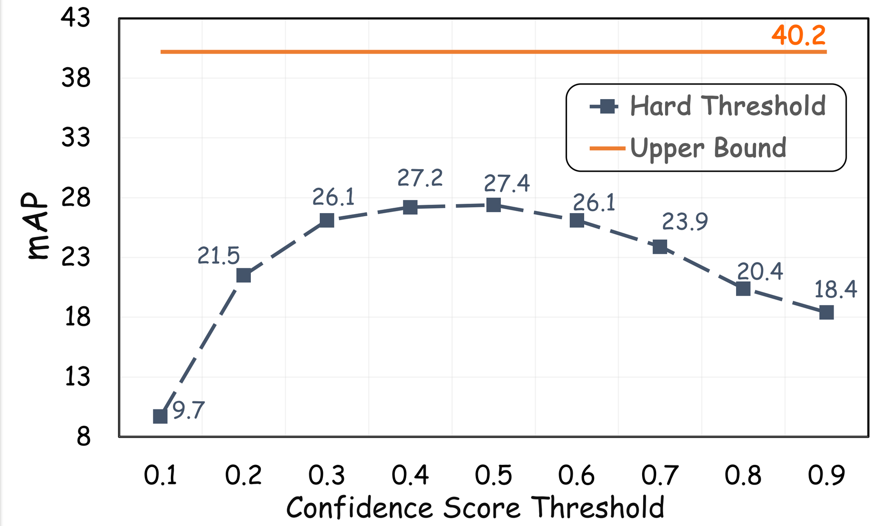
[1]University of Science and Technology of China [2]University of Edinburgh [3]Peking University [4]King Abdullah University of Science and Technology

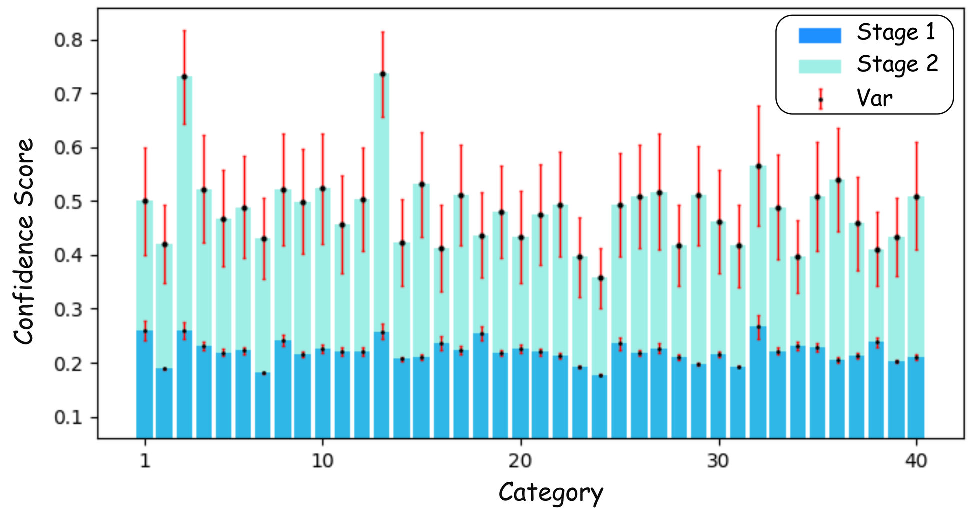
Task

Incremental Object Detection (IOD) aims to overcome catastrophic forgetting while expanding the detector's ability to recognize new classes incrementally.

Motivation

- **Quality Limitation of Pseudo Labels:** The quality of pseudo labels generated by the previous model is constrained by the teacher model's performance, which may introduce noisy data and degrade the learning.
- **Confidence biases across different categories:** Using a fixed threshold for filtering pseudo labels across all classes ignores the varying score distributions among different categories.

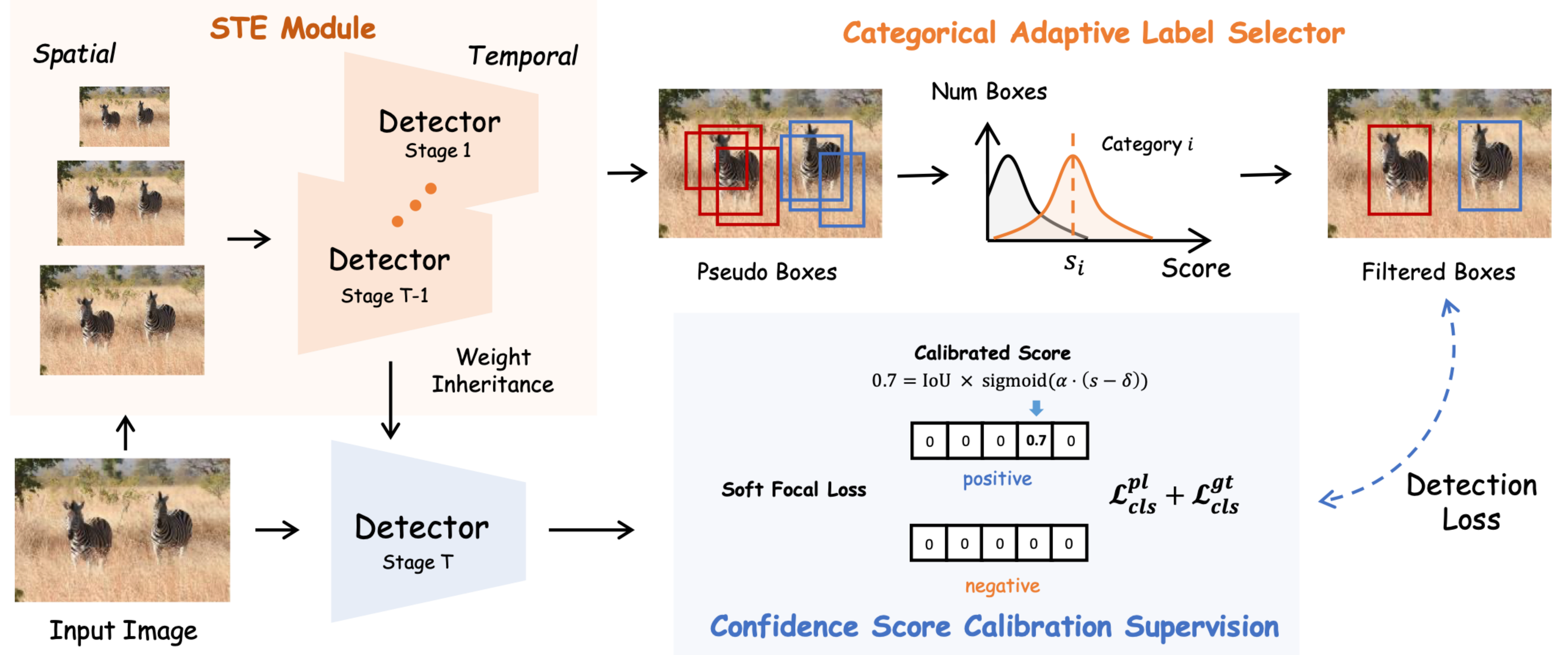




- **Misalignment of Confidence Scores and Localization Quality:** The confidence scores do not linearly correlate with the localization quality of pseudo labels, which can introduce noise if directly used for training.

Methods

The overall framework of PseDet for incremental object detection:



Methods

Spatio-Temporal Enhancement

The Module alleviate the negative effects when learning noisy data from the previous model by reducing spatial noise through multi-scale augmentation and fusion, and mitigating temporal noise accumulation across incremental steps.

- For input image $x \in D$ transformed via A_i, we use the old model Φold to predict pseudo-labels which belong to the old categories:

$$y_i = \Phi_{old}(\mathcal{A}_i(x)),$$

- Then simply reverse them back to the original space and perform unified fusion:

$$\mathcal{P} = \mathcal{F}(A_1^{-1}(y_1), \cdots, A_i^{-1}(y_i)),$$

- Then collect the P in the temporal domain:

$$\mathcal{P} = \mathcal{P}^1 \cup \cdots \cup \mathcal{P}^{i-1},$$

Categorical Adaptive Label Selector

It dynamically determines the class-wise filtering threshold with a simple mathematical prior and fast K-Means pre-computation.

Algorithm 1 Pseudo label selection in stage i

Input: candidate pseudo label set $\mathcal{P}_{input}$; k-means input queue length N.
Output: selected pseudo label $\mathcal{P}_{output}$

```
Initialize the confidence score queue Q^c ← {} for each category c ∈ C^{1:i-1}
for (S, C) ∈ P_input do
    Add the confidence score to the queue S^c → Q^c
end for
for c ∈ C^{1:i-1} do
    K-means(Q^c) → D_T^c, D_F^c
end for
P_output = D_T^1 ∪ ··· ∪ D_T^{i-1}.
```

- Initialize a queue Q^c for each category C^i;
- For each category in the batch, enqueue the samples S^c;
- For each queue, we apply K-Means to fit the true and false distributions.

Confidence Score Calibration Supervision

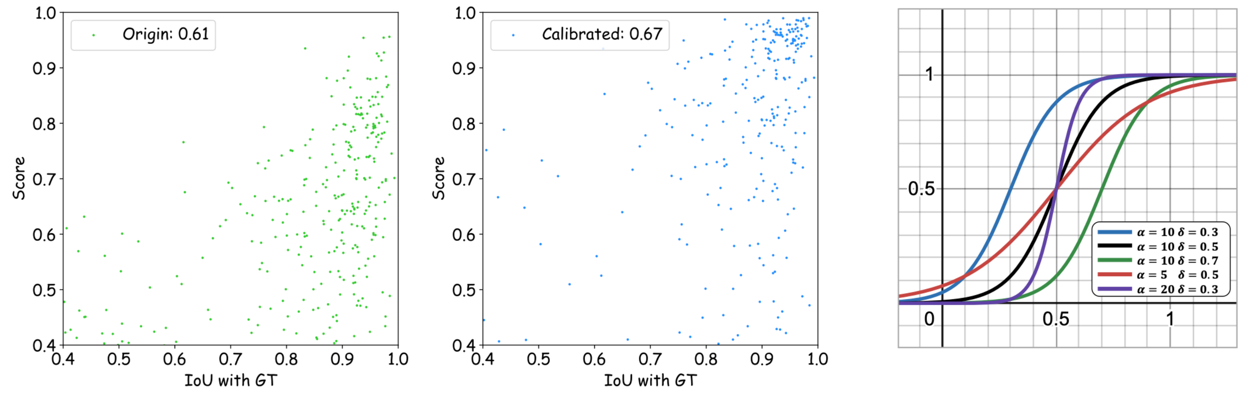
It calibrates the distribution of confidence in order to align the score with the localization quality of the pseudo labels, then integrates the quality into the new-step supervision.

- Map s to q with sigmoid:

$$q(s) = sigmoid(\alpha \cdot (s - \delta)) = \frac{1}{1 + e^{-\alpha \cdot (s-\delta)}},$$

- Use the q_{pl} and the IoU_τ between predictions and matched labels to soften the categorical labels:

$$\hat{y}_{pl} = \tau \cdot q_{pl},$$



(a) Original Distribution (b) Distribution After Mapping (c) Mapping Function Curve

- The class loss for the pseudo labels is: $\mathcal{L}_{cls}^{pl} = -|y - \hat{y}_{pl}|^\beta \left((1-y)\log(1-\hat{y}_{pl}) + y\log(\hat{y}_{pl})\right),$
- The ground truth is universally recognized as high quality $q_{gt} = 1$, we can unify the class loss : $\mathcal{L}_{cls} = \mathcal{L}_{cls}^{pl} + \mathcal{L}_{cls}^{gt} = -|y - \hat{y}|^\beta \left((1-y)\log(1-\hat{y}) + y\log(\hat{y})\right).$

Experiments

One-Step Results

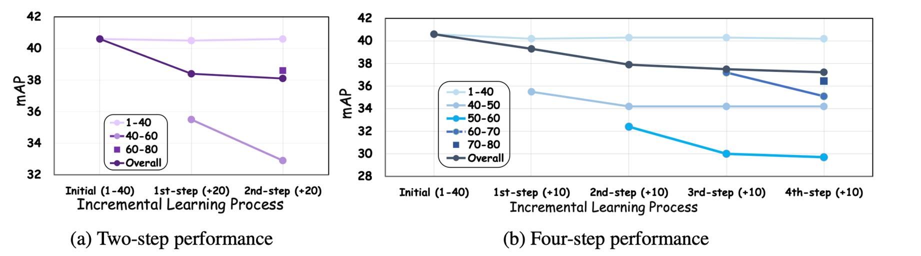
Table 1: **Incremental results on COCO benchmark under the one-step setting**. Most experimental results are borrowed from SDDGR (Kim et al., 2024). AP, AP_{50}, and AP_{75} reflect the overall performance (higher is better) of the model after one step of learning. AbsGap (lower is better) and RelGap (lower is better) represents the absolute gap and the relative gap toward upper bound. The best performance is highlighted in bold.

Scenarios	Method	Detector	$AP\uparrow$	$AP_{50}\uparrow$	$AP_{75}\uparrow$	AbsGap↓	RelGap↓
Upper Bound (Full data)	-	GFL	40.2	58.3	43.6	-	-
		Deformable DETR	47.0	66.1	50.9		
40 + 40	LwF (Li & Hoiem, 2017)	GFL	17.2	25.4	18.6	23.0	57.2%
	RILOD (Li et al., 2019)	GFL	29.9	45.0	32.0	10.3	25.6%
	SID (Peng et al., 2021)	GFL	34.0	51.4	36.3	6.2	15.4%
	ERD (Feng et al., 2022)	GFL	36.9	54.5	39.6	3.3	8.2%
	PseDet (Ours)	GFL	**38.5**	**54.9**	**41.9**	**1.7**	**4.2%**
	CL-DETR (Liu et al., 2023a)	Deformable DETR	42.0	60.1	45.9	5.0	10.6%
	SDDGR (Kim et al., 2024)	Deformable DETR	43.0	62.1	47.1	4.0	8.5%
	PseDet (Ours)	Deformable DETR	**43.5**	**61.5**	**47.2**	**3.5**	**7.4%**
70 + 10	LwF (Li & Hoiem, 2017)	GFL	7.1	12.4	7.0	33.1	82.3
	RILOD (Li et al., 2019)	GFL	24.5	37.9	25.7	15.7	39.1
	SID (Peng et al., 2021)	GFL	32.8	49.0	35.0	7.4	18.4
	ERD (Feng et al., 2022)	GFL	34.9	51.9	37.4	5.3	13.2
	PseDet (Ours)	GFL	**39.2**	**55.6**	**42.8**	**1.0**	**2.5%**
	CL-DETR (Liu et al., 2023a)	Deformable DETR	40.4	58.0	43.9	6.6	14.0%
	SDDGR (Kim et al., 2024)	Deformable DETR	40.9	59.5	44.8	6.1	13.0%
	PseDet (Ours)	Deformable DETR	**44.7**	**62.9**	**48.6**	**2.3**	**4.9%**

Multi-Step Results

Table 2: **Incremental results (AP, %) on COCO benchmark under the multi-step setting**. In the first step, normal training is conducted with 40 categories, followed by the addition of 20 and 10 new categories in the 2-step and 4-step settings each time, respectively.

Method	Detector	40+10+10+10+10				40+20+20	
		(40-50)	(50-60)	(60-70)	(70-80)	(40-60)	(60-80)
RILOD (Li et al., 2019)	GFL	25.4	11.2	10.5	8.4	27.8	15.8
SID (Peng et al., 2021)	GFL	34.6	24.1	14.6	12.6	34.0	23.8
ERD (Feng et al., 2022)	GFL	36.4	30.8	26.2	20.7	36.7	32.4
PseDet (Ours)	GFL	**39.3**	**37.9**	**37.5**	**37.1**	**38.4**	**38.1**
CL-DETR (Liu et al., 2023a)	Deformable DETR	-	-	-	28.1	-	35.3
SDDGR (Kim et al., 2024)	Deformable DETR	42.3	40.6	40.0	36.8	**42.5**	41.1
PseDet (Ours)	Deformable DETR	**42.7**	**41.1**	**41.5**	**41.2**	42.3	**42.8**



(a) Two-step performance (b) Four-step performance

Figure 5: **The performance of our model in the multi-step scenario**. Overall indicates an evaluation of the entire set in the current step, while others refer to evaluations of subsets C^i.

Ablations

Table 3: **Ablation study using the COCO benchmark of 40 classes + 40 classes**. All categories and Old categories represent the performance ($AP/AP_{50}/AP_{75}$, higher is better) evaluated by the model after completing one-step of learning on all 80 categories and the old 40 categories, respectively. The Forgetting Percentage Point (FPP) reflects the performance gap on the initial 40 categories between the start and completion of training, indicating the degree of forgetting with lower values preferred.

Method	All categories ↑			Old categories ↑			FPP ↓		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Fine-tuning	17.9	26.9	19.3	0.0	0.0	0.0	40.6	59.0	44.1
+ Normal Pseudo Labeling	22.8	33.1	24.8	26.6	37.9	29.4	14.0	21.1	14.7
++ Spatial Enhancement	29.9	42.8	32.6	31.9	44.5	35.3	8.7	14.5	8.8
+++ Categorical Adaptive Label Selector	34.1	49.2	37.2	35.9	51.2	39.5	4.7	7.8	4.6
++++ Confidence Score Calibration	**38.5**	**54.8**	**41.9**	**40.8**	**57.5**	**45.1**	**-0.2**	**1.5**	**-1.0**

Analysis

Table 4: The performance of detectors with different queue length of Categorical Adaptive Label Selector under the scenario of 40+40.

Queue Length	All categories			Old categories			New categories		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	$AP_{.5}$	$AP_{.75}$
100	38.5	54.8	41.9	40.8	57.5	45.1	36.2	52.1	38.7
150	38.5	54.9	41.8	40.9	58.0	45.6	36.1	51.8	38.1
200	38.5	54.9	41.8	40.9	57.7	45.2	36.1	52.0	38.5

Table 5: The performance of detectors with varying α and δ of the mapping function $sigmoid(\alpha \cdot (s - \delta))$ under the scenario of 40+40.

α	δ	All categories AP	AP_{50}	AP_{75}	Old categories AP	AP_{50}	AP_{75}	New categories AP	$AP_{.5}$	$AP_{.75}$
10	0.6	37.1	53.1	40.3	39.2	55.9	43.2	34.9	50.3	37.4
10	0.5	**38.5**	**54.8**	**41.9**	**40.8**	**57.5**	**45.1**	**36.2**	**52.1**	**38.7**
10	0.4	37.9	54.2	40.9	40.3	57.2	44.4	35.4	51.3	37.5
5	0.5	38.3	54.5	41.5	40.7	57.3	44.9	35.9	51.7	38.1
20	0.5	38.1	54.6	41.6	40.6	57.4	44.7	35.9	51.8	38.4

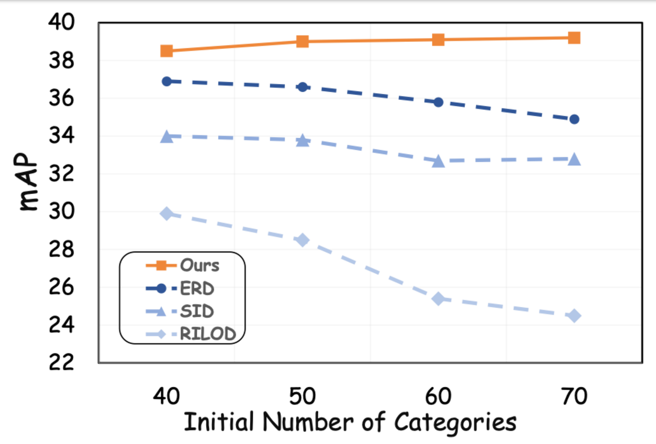


Figure A1: The performance under the one-step scenario. With other methods, the larger the initial set, the faster the forgetting; our method performs better and is suitable for continual learning.