SPATIALITY-AWARE PROMPT TUNING FOR FEW-SHOT SMALL OBJECT DETECTION

SUPPLEMENTARY MATERIAL

A. VERBALIZERS

We provide a detailed description of the verbalizer used in our proposed method, including its definition and the distribution of labels.

A.1. Verbalizers definition

Verbalizers for bounding box size. For verbalizing the major size of bounding boxes in each image, we use four words "tiny", "small", "medium" and "large". These words are masked with the special token [MASK-S]. Each bounding box within an image is classified based on Table 8, where r is the area ratio of the bounding box size to the image size. The label for each image is determined by the predominant category of all bounding boxes contained within that image.

Table 8. Words for verbalizing the size of bounding box. r is the area ratio of the bounding box size to the image size.

Word	Range	
	SODA-D	SmallCOCO
large	$0.05 < r$	$0.3 < r$
medium	$0.02 < r \leq 0.05$	$0.12 < r \leq 0.3$
small	$0.01 < r \leq 0.02$	$0.06 < r \leq 0.12$
tiny	$r \leq 0.01$	$r \leq 0.06$

Verbalizers for bounding box count. For verbalizing counts, we use three words "many", "several", and "few". These words are masked with the special token [MASK-C]. Words for the ground-truth expressions are chosen based on Table 9, where c is the number of ground-truth bounding boxes.

Table 9. Words for verbalizing the number of bounding boxes. Given the ground-truth number of bounding boxes, one of three words is chosen for each image.

Word	Range
many	$20 < c$
some	$10 < c \leq 20$
few	$c \leq 10$

Verbalizers for bounding box position. For verbalizing the major positions of bounding boxes, we employ five descriptive words: "center", "left", "right", "top", and "bottom." Each word is associated with specific regions of an image: center is used when a bounding box lies within the middle third of both the x and y axes; top and bottom correspond to the upper and lower thirds of the y-axis, respectively, combined with the middle third of the x-axis; left and right refer to the leftmost and rightmost thirds of the x-axis. These position words are masked with the special token [MASK-P]. To determine the ground-truth expressions for an image, we count the number of bounding boxes in each defined region as shown in Figure 6, selecting the word that corresponds to the most frequent region for each image.

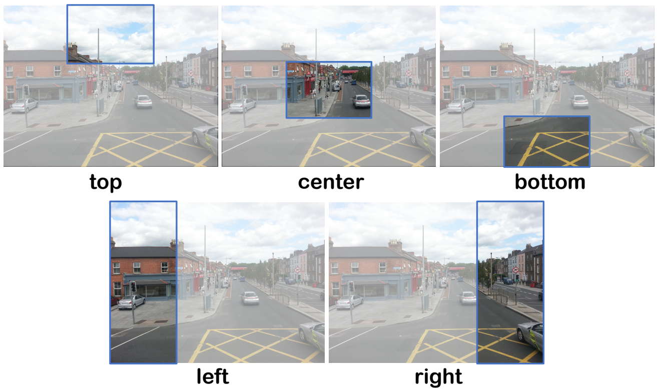

Fig. 6. Words for verbalizing major position of bounding boxes. The ground-truth bounding boxes are counted for each region and the word corresponding to the region with the most frequent bounding boxes is chosen. The size of each region is one-third of the image size.

A.2. Verbalizer label distribution

When extracting verbalization information following the definitions in A.1, the distribution of labels in each dataset is as shown in Figure 7. Compared to the SmallCOCOC dataset, the SODA-D dataset has a relatively higher number of boxes per image, with the object locations being biased towards the center.

B. IMPLEMENTATION AND EXPERIMENT DETAILS

The implementation of the proposed method and comparative methods, such as CoOp [24] and VPT [25], utilized implementation of CFINet [3][1] based on MMDetection, an open-source toolbox for object detection. We conducted the experiments using NVIDIA V100 GPUs for all experiments.

Hyperparameters. The hyperparamers with respect to the optimizer we used are listed in Table 10. Our model adopts the hyperparameter settings from the GroundingDINO [11], applying consistent configurations across both few-shot and full-shot learning scenarios. In the full-shot learning scenario, we adhere to the GroundingDINO's learning rate schedule,

[1] https://github.com/shaunyuan22/CFINet

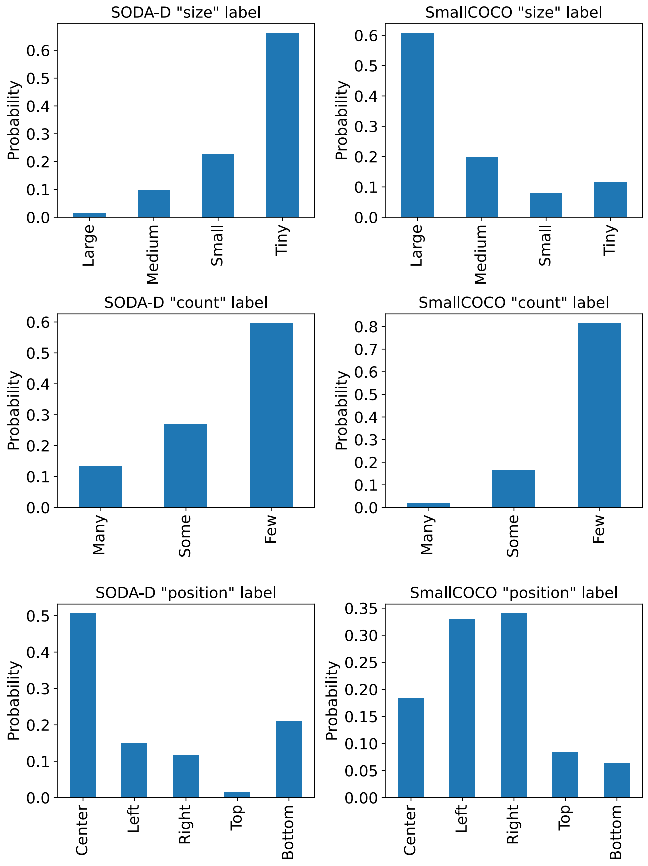

Fig. 7. Label distribution of verbalizers in SODA-D (left) and SmallCOCO (right).

implementing a StepLR schedule with learning rate decay at epoch 11. The optimizer, backbone learning rates, weight decay, and architectural details, including the number and size of encoder and decoder layers, remain aligned with the Grounding DINO specifications, maintaining fidelity to the established framework while addressing the nuances of our specific tasks.

For CoOp, the number of learnable embeddings is set to $M = 16$, following the settings from [24]. For VPT, the number of learnable embeddings is determined experimentally to be $M' = 4$, as this value yielded the best performance on the SODA-D dataset among $M' \in \{1, 2, 4, 8, 16, 32\}$. The number of optional learnable embeddings for SAPT is aligned with VPT, set to $M' = 4$, to maintain consistency in the embedding configuration. For verbalizer (A-C), character-level tokenizer is applied and new embeddings are used for each. For (D), learnable embeddings are appended to the end of the sentence so that the total length becomes 128.

Table 10. Hyperparameters for few-shot and full-shot learning scenarios. The parameters above the separation line apply to few-shot learning, with specific epoch settings for different numbers of shots ($k = 1, 2, 3, 5$ and $k = 10$). The parameters below the separation line are exclusive to the full-shot learning setup, including the learning rate schedule with decay epochs.

Item	Value
optimizer	AdamW
lr	1e-4
lr of image backbone	1e-5
lr of text backbone	1e-5
epochs ($k = 1, 2, 3, 5$)	4
epochs ($k = 10$)	3
weight decay	0.0001
clip max norm	0.1
number of encoder layers	6
number of decoder layers	6
dim feedforward	2048
hidden dim	256
dropout	0.0
nheads	8
number of queries	900
set cost class	1.0
set cost bbox	5.0
set cost giou	2.0
ce loss coef	2.0
bbox loss coef	5.0
giou loss coef	2.0
batchsize	644
running average coefficient β_1	0.9
running average coefficient β_2	0.999
epsilon ϵ	10^{-8}
learning rate schedule	StepLR
learning rate decay	0.1
learning rate decay epoch	11
epochs	12

Architecture. We describe the architecture details of SAPT applied to Grounding DINO. Let $\boldsymbol{T}, \bar{\boldsymbol{T}}, \boldsymbol{X}$ and $\bar{\boldsymbol{X}}$ be a text prompt, vanilla text features, an image prompt and vanilla image features, respectively, given by

$$\boldsymbol{T} = [\, \boldsymbol{T}_{\text{Det}}, \boldsymbol{V} \,], \tag{7}$$

$$\bar{\boldsymbol{T}} = \mathrm{TextBackbone}(\boldsymbol{T}), \tag{8}$$

$$\boldsymbol{X} = [\, \boldsymbol{E}, \boldsymbol{P} \,], \tag{9}$$

$$\bar{\boldsymbol{X}} = \mathrm{ImageBackbone}(\boldsymbol{X}), \tag{10}$$

where $\mathrm{TextBackbone}$ is the frozen BERT model and $\mathrm{ImageBackbone}$ is the frozen Swin Transformer model. Note that following VPT [25], the embeddings of added prompts are ignored during patch merging stages of Swin. Then, assuming the vanilla features can be written as

$$\bar{\boldsymbol{T}} = [\, \bar{\boldsymbol{T}}_{\text{Det}}, \bar{\boldsymbol{V}} \,], \tag{11}$$

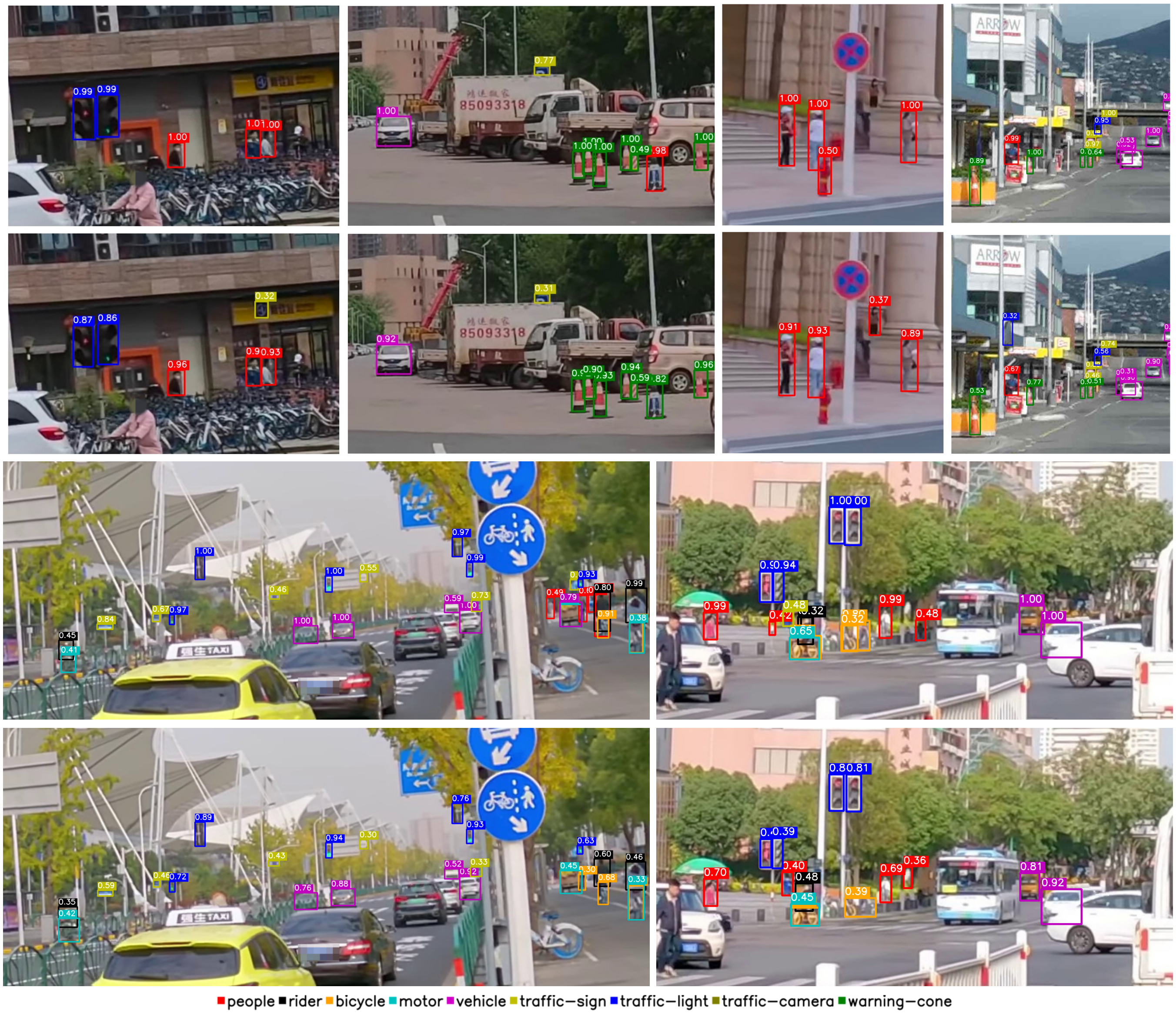
🟥people ■rider 🟧bicycle 🟦motor 🟪vehicle 🟨traffic−sign 🟦traffic−light 🟩traffic−camera 🟩warning−cone

Fig. 8. Qualitative comparison of results between CFINet [3] (top) and proposed SAPT (bottom).

the frozen encoder (feature enhancer) is applied to the features as follows:

$$(\tilde{\boldsymbol{T}}, \tilde{\boldsymbol{X}}) = \text{Encoder}(\bar{\boldsymbol{T}}, \bar{\boldsymbol{X}}). \tag{12}$$

This output can be written as follows:

$$\tilde{\boldsymbol{T}} = [\, \tilde{\boldsymbol{T}}_{\text{Det}}, \tilde{\boldsymbol{V}} \,], \tag{13}$$

$$\tilde{\boldsymbol{X}} = [\, \tilde{\boldsymbol{E}}, \tilde{\boldsymbol{P}} \,]. \tag{14}$$

The verbalizer head is a linear layer. Given a set of token indexes of special mask tokens $\mathcal{M}$ ($|\mathcal{M}| = 3$ for the three masks), linear layers for each $i \in \mathcal{M}$ is applied to predict masked words:

$$\hat{\boldsymbol{y}}_i = f_i(\hat{\boldsymbol{v}}_i). \tag{15}$$

to which the masked prediction loss is applied as in Eq. (9).

C. DATASET DETAILS

In this research, we constructed k-shot datasets to evaluate the performance of few-shot learning in small object detection, due to the lack of existing benchmarks for few-shot small object detection. The k-shot dataset used in the experiments was constructed by randomly sampling k images from each category that contain annotations relevant to that category, and then integrating them all together.

In addition, to demonstrate the effectiveness of our approach beyond the SODA-D dataset, we constructed the SmallCOCO dataset, a subset of the COCO dataset focused

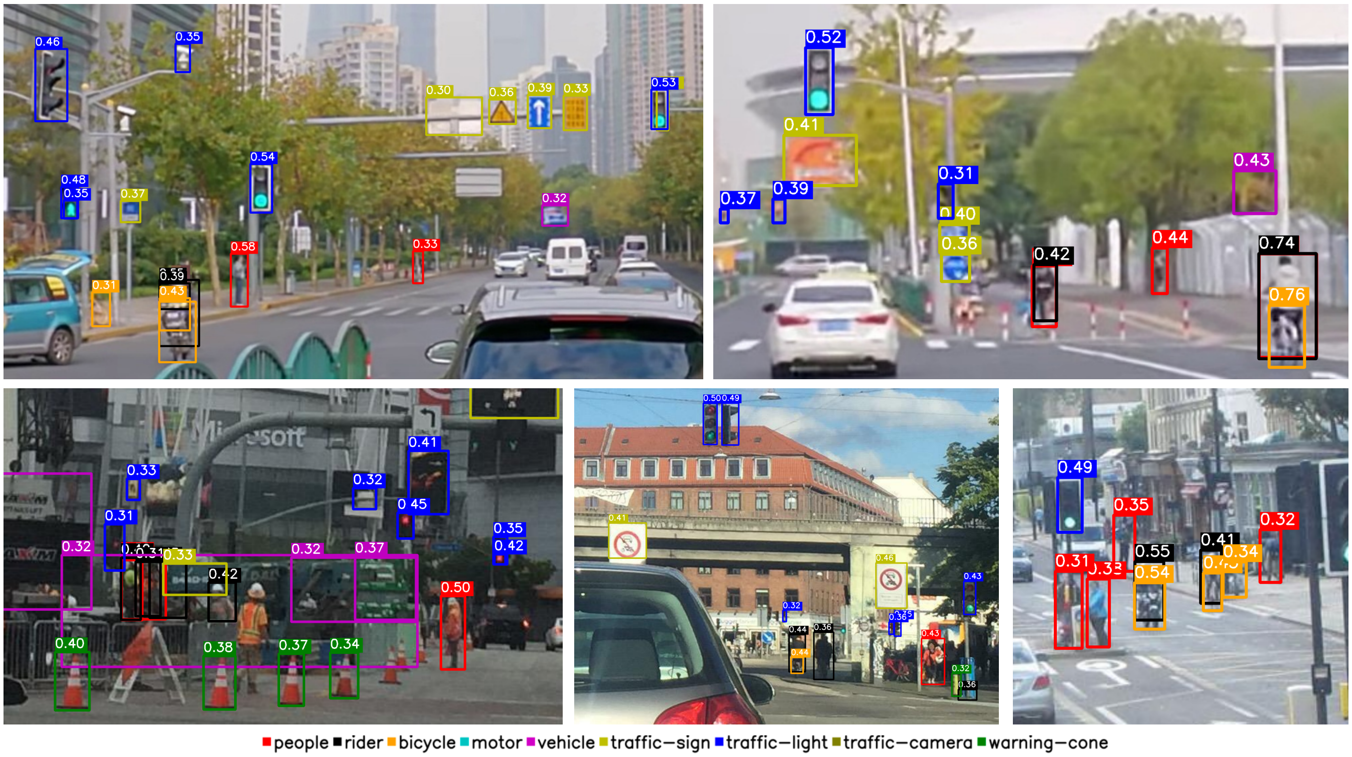
🟥people ■rider 🟧bicycle ■motor 🟪vehicle 🟨traffic−sign 🟦traffic−light ■traffic−camera 🟩warning−cone

Fig. 9. Visualizations of our SAPT predictions trained in one-shot learning scenario.

Table 11. Categories included in the SmallCOCO dataset

Id	Super Category	Category
1	person	person
3	vehicle	car
9	vehicle	boat
10	outdoor	traffic light
16	animal	bird
31	accessory	handbag
44	kitchen	bottle
47	kitchen	cup
62	furniture	chair
84	indoor	book

on categories containing a large number of small objects. Specifically, following the evaluation criteria of SODA-D, we targeted annotations with an area size of 2000 pixels or less, counted the number of annotations, and sorted them in descending order to narrow down to 10 categories. As a result, the categories included in the SmallCOCO dataset are as shown in Table 11. Furthermore, after narrowing down the categories, similar to the SODA-D dataset, annotations larger than 2000 pixels were replaced with an "ignore" class, ensuring that the evaluation is appropriately conducted only on small objects.

D. MORE RESULTS AND ANALYSIS

D.1. Qualitative comparison

We present a qualitative comparison of our proposed SAPT method with the current state-of-the-art, CFINet [3], as shown in Figure 8. For CFINet, we used a model that was retrained in our own environment to ensure the integrity of the results. Both methods demonstrate impressive detection capabilities across a wide array of categories, evidencing their proficiency in managing diverse object classes. A detailed analysis of the visualized results reveals a marginal yet noticeable superiority of SAPT, especially in reducing classification errors, false positives, and missed detections - areas where CFINet, while generally effective, shows occasional limitations. Furthermore, the visualization from both methods underscores the difficulty in differentiating between similar categories in the SODA-D dataset, notably between "motor" and "bicycle", and "people" and "rider". These observations not only accentuate the intricacies of fine-grained categorization but also suggest avenues for enhancement in future object detection algorithm developments.

D.2. Class-wise results comparison

Table 12 provides a comparative analysis between the state-of-the-art CFINet method [3] and our proposed SAPT approach, detailing their performance across various categories. For each metric, superior results are emphasized in bold.

Table 12. Class-wise average precision (AP) results of CFINet [3] and SAPT (outs) on SODA-D dataset (Better results in bold).

Class	Method	AP	AP_{50}	AP_{75}	AP_{eS}	AP_{rS}	AP_{gS}	AP_{N}
people	CFINet	37.7	66.8	35.6	**11.4**	30.2	43.5	53.8
	SAPT	**40.6**	**72.1**	**38.9**	10.5	**32.3**	**47.1**	**59.8**
rider	CFINet	18.5	49.6	9.3	7.5	14.7	23.2	31.9
	SAPT	**22.1**	**58.2**	**12.5**	**8.6**	**19.7**	**27.0**	**33.7**
bicycle	CFINet	13.7	33.0	8.7	3.1	**10.1**	16.7	28.4
	SAPT	**15.5**	**36.7**	**11.3**	**3.3**	8.6	**19.9**	**37.4**
motor	CFINet	25.6	61.4	16.1	13.7	25.4	29.0	33.3
	SAPT	**29.0**	**65.6**	**18.8**	**14.8**	**27.2**	**34.0**	**37.4**
vehicle	CFINet	46.4	80.4	47.2	**23.6**	**41.1**	53.7	65.9
	SAPT	**47.8**	**81.9**	**49.2**	21.3	41.0	**56.9**	**70.4**
traffic-sign	CFINet	46.6	76.8	50.3	**24.1**	42.3	54.1	63.9
	SAPT	**47.8**	**78.8**	**51.8**	24.1	**42.4**	**56.4**	**66.8**
traffic-light	CFINet	38.5	74.0	35.3	**24.2**	37.2	45.1	54.0
	SAPT	**40.1**	**78.7**	**36.6**	23.8	**37.9**	**48.4**	**57.6**
traffic-camera	**CFINet**	**15.2**	**34.9**	**10.0**	**7.5**	**15.9**	**21.8**	28.7
	SAPT	12.5	29.1	8.5	5.8	13.0	18.9	**29.7**
warning-cone	CFINet	31.6	65.5	26.1	13.1	28.5	38.3	47.5
	SAPT	**34.8**	**73.0**	**27.8**	**16.7**	**30.7**	**42.4**	**52.1**

Table 13. Class-wise average precision (AP) resuls of one-shot learning with SAPT (validation set).

Class	AP	AP_{50}	AP_{75}	AP_{eS}	AP_{rS}	AP_{gS}	AP_{N}
people	17.1	35.4	13.7	2.6	9.3	22.2	33.6
rider	1.8	6.0	0.7	0.0	1.3	2.6	4.7
bicycle	7.8	22.4	3.6	0.1	3.3	11.2	19.6
motor	0.1	0.3	0.1	0.0	0.0	0.2	0.4
vehicle	15.9	32.3	14.2	5.3	13.4	19.9	24.4
traffic-sign	22.2	41.4	21.5	7.8	17.0	29.4	36.5
traffic-light	25.1	52.5	21.0	10.8	23.5	33.1	43.2
traffic-camera	2.5	5.6	1.7	0.7	2.7	4.8	7.2
warning-cone	7.4	16.5	5.4	0.3	6.7	9.1	17.5

SAPT, our proposed method, shows significant performance improvements in most classes. While both SAPT and CFINet have limited effectiveness in the "traffic-camera" category, which is considered a challenging domain for detection, CFINet slightly outperforms SAPT in this area. This suggests potential areas for further improvement in our method.

D.3. Detailed results in one-shot learning scenario

We present the visualization of prediction results for the one-shot learning scenario of our proposed SAPT method in Figure 9, along with detailed class-wise results in Table 13. While there is a noticeable degradation in performance compared to the full-shot scenario, it is remarkable that SAPT successfully detects a significant number of categories even in the one-shot setting. Notably, as discerned from the class-wise detailed results in Table 13 and the visualization outcomes, SAPT demonstrates efficient detection in categories like "traffic-sign" and "traffic-light". Additionally, the method shows reasonably good results for "bicycle" and "warning-cone". While the "vehicle" class exhibits relatively good performance in quantitative results, qualitative analysis through visualization suggests that the performance might not be as satisfactory. One reason for this could be the ambiguous class name "vehicle" in the SODA-D dataset to define cars. This ambiguity might not translate well qualitatively, and is considered to have a moderate impact on the effectiveness

Table 14. Detailed result of various verbalizer prompt comparison (Table 6).

	[MC]	[MS]	[MP]	mAP	mAP_{50}	mAP_{75}	mAP_{eS}	mAP_{rS}	mAP_{gS}	mAP_{N}
(A)	✓			10.0	21.5	8.2	2.9	7.9	13.1	18.4
(B)		✓		9.7	20.9	7.9	2.8	7.6	12.8	17.9
(C)			✓	10.2	21.9	8.3	2.9	7.8	13.5	19.0
(D)	✓	✓	✓	11.1	23.6	9.1	3.1	8.6	14.7	20.8

Table 15. Detailed result of effectiveness comparison of learnable embeddings for image prompts (Table 7).

Method	mAP	mAP_{50}	mAP_{75}	mAP_{eS}	mAP_{rS}	mAP_{gS}	mAP_{N}
CoOp	8.4	18.7	6.6	2.4	6.4	11.1	16.0
VPT	8.3	18.9	6.3	2.5	6.5	10.9	15.6
CoOp + VPT	8.2	18.8	6.0	2.4	6.3	10.8	15.5
SAPT w/o optional LE	11.1	23.6	9.1	3.1	8.6	14.7	20.8
SAPT w/ optional LE	11.2	24.0	9.1	3.1	8.7	14.7	20.8

of vision-language detection models. Traditionally, dataset category names have been largely symbolic, but with the growing significance of vision-language models, reassessing the suitability of these category names in datasets becomes increasingly important.

D.4. Detailed results of ablation study

The detailed results of the comparative experiments involving various verbalizer prompts and learnable embeddings for image prompts, as conducted in Subsection 4.4 of the main manuscript, are respectively presented in Table 14 and Table 15. These findings consistently underscore the advantage of using multiple verbalizer masked tokens, among other insights. This aligns with the trends and conclusions reported in the main text.