Supplementary Materials

1. Overview of APLGOS's Training Phases

APLGOS is trained in a three-phase process to mitigate overfitting on ID and OOD data while enhancing the precision of its decision boundary. During the first training phase, APLGOS is pre-trained on ID data and APLGOS only needs to perform detection on ID categories. PLM and TAM do not work at this phase. During the second training phase, APLGOS is trained to perform detection on only ID data, at the same time, PLM and TAM only work on ID data. During the third training phase, APLGOS needs to perform detection on both ID and OOD data, whereas PLM and TAM only work on OOD data.

Phase	PLM		TAM		DM	
	ID	OOD	ID	OOD	ID	OOD
I					✓	
II	✓		✓		✓	
III		✓		✓	✓	✓

Table S1. The overview of APLGOS's three training phases. "✓" denotes ID or OOD data is available for the module. "**I**","**II**","**III**" represent the first, second, and third training phases, respectively. PLM, TAM, and DM stand for Prompt Learning Module, Text-Image Alignment Module, and Decision Module, respectively.

2. More Technical Details

$\mathcal{L}_{align}^{ood}$ and $\mathcal{L}_{loc}^{ood}$ guide the model in separating OOD samples from ID data **(Eq.(10), (11))**. The $\mathcal{L}_{reg}$ supervises the detection boxes and is implemented using the **Smooth L1** loss. The object detector used by the "pre-detecting" module is **Faster R-CNN**, with the Region Proposal Network (RPN) serving as the proposal generator. The coordinate decoder is a two-layer MLP that implicitly encodes the coordinates of ID/OOD regions into prompts for ID/OOD classification. **Standardization** means a controlled generation process using predefined templates and guidelines, ensuring semantic consistency, format compliance, and guided expansion of the text space rather than free paraphrasing.

3. Broader Experimental Comparisons

We include additional comparisons with the most recently identified synthesis-based OOD detection methods — NPOS (ICLR 2023) and ID-like (CVPR 2024) — as shown in Table S2. Since official results on our datasets are unavailable, we reproduce them locally for fair comparison.

Methods	FPR95 ↓	AUROC ↑	AUPR ↑
	OOD: MS-COCO2017 / OpenImages		
NPOS (*ICLR 2023*)	76.34 / 73.03	67.69 / 34.82	88.97 / 68.67
ID-like (*CVPR 2024*)	67.74 / 60.53	82.59 / 86.32	95.81 / 94.43
APLGOS (ResNet50)	47.16 / 49.66	87.89 / 85.91	98.80 / 97.54
APLGOS (RegNetX4.0)	**45.96 / 47.10**	**89.19 / 88.49**	**99.00 / 98.30**

Table S2. Broader Experimental Comparisons. ID dataset: PASCAL VOC; OOD datasets: MS-COCO2017 / OpenImages.

4. Qualitative Analysis

During LLMs' warm start, the main computational overhead comes from the **visual modality**. Excessively large hyperparameters lead to instability in the training process, making convergence difficult.

5. The Way to Introduce Location Information in Prompts

We introduce location information in prompts to indicate the position of the region image within the original input image, assisting the model in aligning image and text embeddings in the hidden space. We compare explicitly introducing location information in prompts with implicitly introducing it. In the explicit method, the <LOC> tokens in the prompts for each in-distribution region image are replaced with the original region coordinates, rather than being obtained through regression. The results are shown in Table S3. The implicit method achieves better detection performance and significantly reduces computational overhead, with GPU memory usage of approximately 22GB for the implicit method compared to 26GB for the explicit method.

6. The Values of Hyperparameters in the Total Loss

After incorporating the classification loss $\mathcal{L}_{cls}$ and the location loss $\mathcal{L}_{loc}$, the total loss can be expressed as:

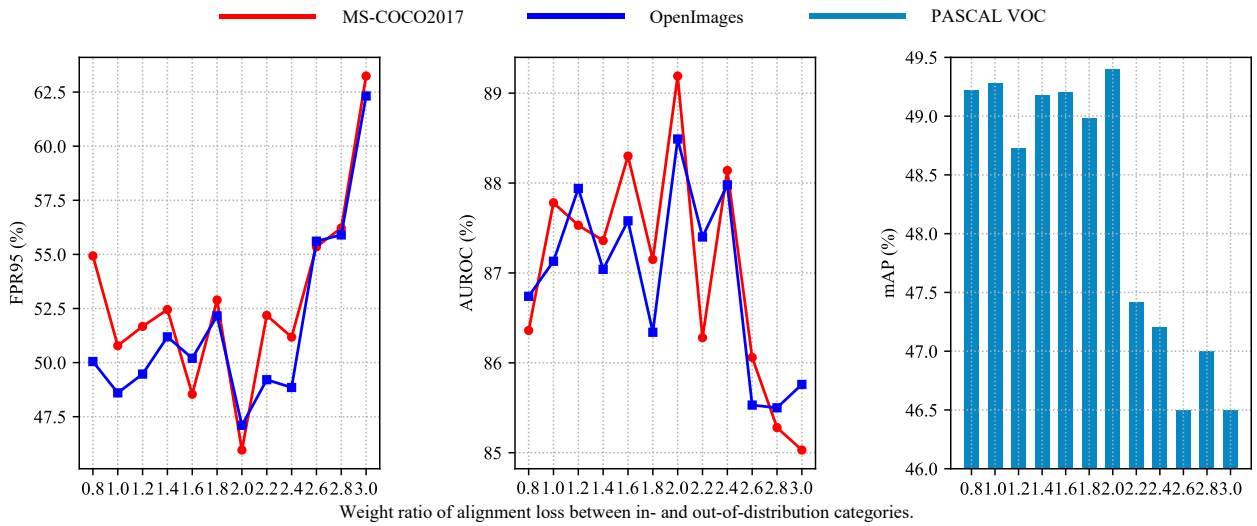

Figure S1. Weight ratio of alignment loss between ID and OOD categories. The horizontal coordinate is the weight ratio $\Gamma_2 = \mathcal{L}_{align}^{id}/\mathcal{L}_{align}^{ood}$. The vertical coordinates from left to right are FPR95, AUROC and mAP. We use PASCAL VOC as ID dataset, MS-COCO2017 and OpenImages as OOD datasets.

Prompts Way	FPR95 ↓	AUROC ↑	mAP (ID) ↑
	OOD: MS-COCO2017 / OpenImages		
Explicitly prompt	51.97 / 55.24	87.27 / 85.89	47.5
Implicitly prompt	**47.16 / 49.66**	**87.89 / 85.91**	**48.8**

Table S3. We conduct ablation experiments on different ways of introducing location information in prompts. We use ResNet50 and a Transformer as the backbone, with PASCAL VOC as the in-distribution (ID) dataset, and MS-COCO2017 and OpenImages as out-of-distribution (OOD) datasets.

Phase	ξ_1	ξ_2	ξ_3	ξ_4	τ	κ
I	0	0	1	1	∞	∞
II	1	1	1	1	1	1
III	1	0	1	1	0	∞

Table S4. The values of hyper-parameters in total loss $\mathcal{L}$ at different training phases. Different values of hyper-parameters determine which loss works in current training phase. "∞" indicates that the value of this hyperparameter in the current training phase can be arbitrary, as it does not play any role.

$$\begin{aligned}\mathcal{L} =& \xi_1[\gamma_1\tau\mathcal{L}_{align}^{id} + \gamma_2(1-\tau)\mathcal{L}_{align}^{ood}] \\ &+ \gamma_3\xi_2[\kappa\mathcal{L}_{loc}^{id} + (1-\kappa)\mathcal{L}_{loc}^{ood}] \\ &+ \gamma_4\xi_3\mathcal{L}_{cls} + \gamma_5\xi_4\mathcal{L}_{reg} + \overline{\mathcal{W}}.\end{aligned} \tag{1}$$

γ_1,γ_2,γ_3,γ_4,γ_5 are the hyper-parameters, ξ,τ,κ determine the loss functions used in the current training phase and $\xi_i = \{0, 1\}$, $\tau = \{0, 1\}$. The values of these hyperparameters at different training phases are shown in Table S4.

7. Weight Ratio of Alignment Loss for ID and OOD Categories

To investigate the effect of the constraint strength of ID and OOD alignment loss on the model's performance, we conduct extensive ablation experiments on the weight ratio $\mathcal{L}_{align}^{id}/\mathcal{L}_{align}^{ood}$ between $\mathcal{L}_{align}^{id}$ and $\mathcal{L}_{align}^{ood}$ in total loss $\mathcal{L}$. The results are shown in Figure S1. When the weight ratio $\mathcal{L}_{align}^{id}/\mathcal{L}_{align}^{ood}$ is set to 2, the model achieves the best performance. However, as $\mathcal{L}_{align}^{id}/\mathcal{L}_{align}^{ood}$ continues to increase, performance degrades significantly. This is because the ID alignment loss $\mathcal{L}_{align}^{id}$ imposes excessive constraints on the model, leading to overfitting on ID categories and preventing the model from learning effective decision boundaries.

8. Different Types and Versions of Vision-Language Models

As shown in Table S5, we evaluate the robustness of APLGOS using both open-source (*i.e.* Qwen2.5-7B) and closed-source (*i.e.* GPT4o and GPT3.5) LLMs as standardization models, demonstrating that APLGOS maintains robustness in the expanded textual space.

9. Visualization of Detection Results

Figure S2 shows the detection results of our proposed APLGOS on the ID dataset. Here, we use PASCAL VOC as the ID dataset, with a RegNetX4.0-based image encoder and a Transformer-based text encoder. This also demonstrates that our APLGOS achieves strong performance not only on

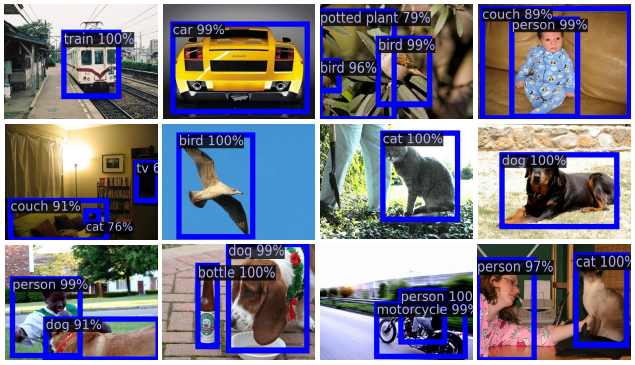

Figure S2. The detection results of our proposed APLGOS on ID dataset. Here we use PASCAL VOC as ID dataset, and we use RegNetX4.0-based image encoder and Trasformer-based text encoder. Our APLGOS can accurately detect all objects with high confidence score.

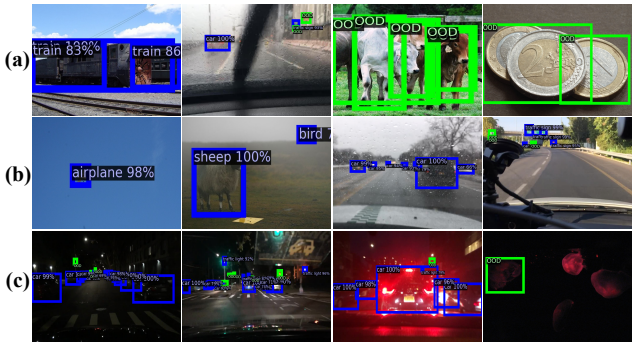

Figure S3. More visualization results under scenarios including (a) occlusion, (b) Long-distance targets and (c) lighting variation.

Methods	FPR95 ↓	AUROC ↑	mAP (ID) ↑
	OOD: MS-COCO2017		
Qwen2.5-7B*	47.30	87.25	98.73
GPT-4o	47.27	87.56	98.73
GPT-3.5 (APLGOS)	47.16	87.89	98.80

Table S5. Ablation studies of the different expanded textual space. "*" indicates that the model is deployed locally.

OOD categories but also on ID categories. Figure S3 shows strong robustness in detecting under **(a) more occlusion**, **(b) long-distance targets**, and **(c) lighting variation** scenarios of APLGOS. However, in dimly lit scenes, there may be missed or false detections.

10. Algorithm for Calculating Similarity Scores

The calculation of the Algorithm for Calculating Similarity Scores runs through the three training stages of APLGOS and is used to calculate the alignment loss for aligning the visual and text modalities through contrastive learning. We summarize its detailed calculation process in Algorithm 1.

Algorithm 1: Similarity score **S** calculation

Given notations
  $\mathbb{S}$: ChatGPT standarizing operation
  $\mathbb{C}$: Similarity score calculating operation
  $\mathbb{P}$: Sample from class-conditional gaussian distribution
  $\mathbb{F}$: Determine the number of iterations based on dataset
Input
  $[\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_b]$: ID detected images of each RGB image
Output
  S: Similarity score
Initial setting
  $\mathcal{Q}_T \leftarrow \varnothing, \mathcal{Q}_I \leftarrow \varnothing, \mathcal{E}_1, \mathcal{E}_2 \leftarrow \mathbb{F}(Dataset)$
  Phase I - Pre-training for ID categories
while $\mathcal{Q}_I$ *is NOT FULL* **do**
  Add $\mathbf{X}_i$ to $\mathcal{Q}_I$
  Classify and Regress on $[\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_b]$
end
  Phase II - Prompt learning for ID categories
foreach *iter in* $\mathcal{E}_1$ **do**
  $\hat{\mathbf{T}}_i \leftarrow \mathbb{S}(\mathrm{Q}_i^{\mathrm{A}}, \mathrm{M}, \mathrm{G}_0), \hat{\mathbf{T}} = [\hat{\mathbf{T}}_1, \hat{\mathbf{T}}_2, ..., \hat{\mathbf{T}}_b]$
  if $\mathcal{Q}_T$ *is NOT FULL* **then**
    Add $\hat{\mathbf{T}}_i$ to $\mathcal{Q}_T$
  else
    Move $\mathcal{Q}_{T\,i}$ to $\mathcal{Q}_{T\,i-1}, (2 \leq i \leq |\mathcal{Q}_T|)$
    Add $\hat{\mathbf{T}}_i$ to $\mathcal{Q}_T$
  end
  $\mathbf{S} \leftarrow \mathbb{C}(\hat{\mathbf{X}}, \hat{\mathbf{T}})$
end
  Phase III - Prompt learning for OOD categories
foreach *iter in* $\mathcal{E}_2$ **do**
  Calculate $\hat{\mu}_i, \hat{\nu}_i$ and $\hat{\sigma}, \hat{\eta}$ using Eq. 4 and Eq. 5
  Find $\mathcal{V}, \mathcal{U}$ using Eq. 6-8
  $\hat{\mathbf{X}}^{\dagger} \leftarrow \mathbb{P}(\mathcal{V}), \hat{\mathbf{T}}^{\dagger} \leftarrow \mathbb{P}(\mathcal{U})$
  $\mathbf{S} \leftarrow \mathbb{C}(\hat{\mathbf{X}}^{\dagger}, \hat{\mathbf{T}}^{\dagger})$
end