

ITC 2/54 Information Technology and Control Vol. 54 / No. 2 / 2025 pp. 712-734 DOI 10.5755/j01.itc.54.2.41078	Enhancing Open-Set Few-Shot Object Detection with Limited Visual Prompts	
	Received 2025/04/03	Accepted after revision 2025/05/09
	HOW TO CITE: Yang, Q., Tian, Y., Sun, J., He, F. (2025). Enhancing Open-Set Few-Shot Object Detection with Limited Visual Prompts. <i>Information Technology and Control</i> , 54(2), 712-734. https://doi.org/10.5755/j01.itc.54.2.41078	

Enhancing Open-Set Few-Shot Object Detection with Limited Visual Prompts

Qinghua Yang, Yan Tian

School of Artificial Intelligence, China University of Mining and Technology (Beijing);
e-mails: snowicelean@163.com; 13681362249@163.com

Jing Sun*

School of Basic Education, Beijing Polytechnic College, Beijing, China; e-mail: sjing@bgy.edu.cn

Fangyuan He

College of Applied Science and Technology of Beijing Union University, Beijing, China;
e-mail: ykftangyuan@buu.edu.cn

Corresponding author: sjing@bgy.edu.cn

The text-prompt-based open-vocabulary object detection model effectively encapsulates the abstract concepts of common objects, thereby overcoming the limitations of pre-trained models, which are restricted to detecting a fixed, predefined set of categories. However, due to data scarcity and the constraints of textual descriptions, representing rare or complex objects solely through text remains challenging. In this study, we propose an open-set detection model that supports both visual and textual prompt queries (VTP-OD) to enhance few-shot object detection. A small number of visual prompts not only provide rich class-wise visual features, which enhance class textual representations, but also enable flexible extension to new classes for different downstream tasks. Specifically, we incorporate two adaptation modules based on cross-attention to adapt the pre-trained vision-language model, allowing it to support both text and visual queries. These modules facilitate (i) visual fusion between a limited number of visual prompts and query images and (ii) visual-language fusion between class-aware visual features and textual representations of the classes. Subsequently, the model undergoes prompt tuning using the available few-shot downstream data to adapt to target detection tasks. Experimental results demonstrate that our model outperforms the pre-trained model on the LVIS and COCO benchmarks. Furthermore, we validate its effectiveness on the real-world CoalMine dataset.

KEYWORDS: Object Detection, Open-Set, Few-Shot, Vision-Language, Coalmine

1. Introduction

Object detection is a fundamental task in computer vision that has significantly advanced over the past decade thanks to the research on deep learning [10, 15, 31, 36]. Conventional deep learning-based object detection methods typically rely on large-scale annotated datasets to achieve strong performance. However, in real-world applications, data scarcity is a common challenge. Few-shot object detection (FSOD) has been proposed as a promising approach to address this issue by enabling object detection with limited labeled samples.

Previous works on FSOD provide few exemplar images of novel classes to mimic baby learning and adapts the model trained on base classes to novel classes. However, these studies often follow a fixed dataset split—such as partitioning PASCAL VOC or COCO into base and novel classes—training on the base classes and evaluating on the novel ones. This setup poses significant limitations in terms of both practical applicability and generalization performance.

Recently, combining large-scale pre-trained models with few-shot learning techniques has become a key research direction, enabling rapid adaptation to various downstream tasks. In this work, we investigate how to leverage a pre-trained VL detection model and adapt it to downstream object detection tasks using only a few samples from the target domain.

Following the rapid progress of vision language pre-training, OVD [48] has been proposed to address the problem of detecting objects from novel categories beyond the base categories during training. Because the conventional detectors are trained and evaluated on a fixed, closed set that contains only a limited number of predefined categories, which significantly limits the model's application in complex real-world environments [20,26]. This new paradigm of object detection takes advantage of the generalization and flexibility of language and has gained increasing attention from the community [7, 12, 19, 23, 37]. Previous studies such as GLIP [19] and Grounding-DINO1.5 have cross-modality deep feature fusion to align features of different modalities. Pre-training the model on large-scale datasets, such as Objects365 [34], GoldG (0.8M human-annotated gold grounding data have been curated by MDETR [12]), and Cap4M, often results in better performance. While these

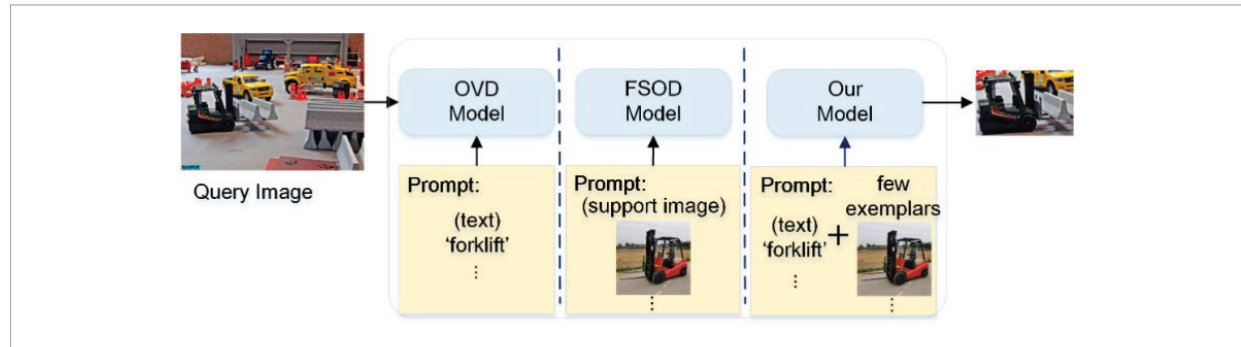
works deliver remarkable performance on OVD tasks, they are computationally expensive and time-consuming. Some studies use knowledge from a pre-trained vision-language (VL) model, such as CLIP [30], for knowledge distillation [7, 52, 37] or prompt learning [5,14,41]. However, these works only used split LVIS [8] and COCO [22] datasets for evaluation, lacking validation for different downstream datasets in the real world. Therefore, we use the deeply integrated pre-trained VL model, GLIP, as the base model.

In general, open-vocabulary object detection queries the objects of interest in images using text. However, the text-only methods may not be the best for the following reasons: First, text-only may lead to concept ambiguity problems in case of homonyms, e.g., 'letter' may refer to either 'mail' or 'character'. Second, users may only be familiar with the main categories of certain rare objects, such as 'flower' versus 'datura flower', which could introduce additional descriptive information. Third, the text may struggle to describe the difference in detail. Since a picture can convey a lot of information, it can effectively illustrate the complex visual details of an object. Therefore, we explore a method that combines text and images while querying the images. Inspired by the meta-based learning mechanism of few-shot learning [9, 13, 25, 28, 38, 39, 40], where query images are combined with few support images to create episodes during training, we adopt a similar mode to couple query features with support features. The work in [18] also adopts a similar approach, which introduces a universal visual in-context prompting framework that leverages visual context to understand new categories, but the model only used visual unimodality. As described in devit [50], both open-vocabulary and few-shot learning belong to the open set, with the primary difference being their category representations: language and images, respectively.

In this paper, by combining OVD and FSOD methods, we propose an open-set detection model that supports simultaneous querying through both visual and textual prompts, using a small number of reference images. As shown in Figure 1, our model's query prompts differ from those of vanilla OVD and FSOD approaches. It takes advantage of both the fine-grained information contained in images and the stronger generalization ability of text, which complement each other and col-

Figure 1

Comparison of query prompts across different methods. OVD relies solely on textual prompt, FSOD performs queries based on support images, whereas our model utilizes both textual and visual prompts simultaneously.



lectively enhance the model's generalizability. Based on the frozen pre-trained GLIP model, we insert adaptation modules and train only the new modules on the Object365 dataset, thus avoiding the heavy training burden associated with training from scratch.

Specifically, objects in the training dataset images are cropped to obtain usable visual prompts. During training, at each iteration, k visual prompts are randomly selected for each class, and self-attention is applied to weight them, assigning different weights to the prompt features based on their semantic information. Then, a masked multi-head cross-attention (MHCA) is applied to fuse the query image and visual prompt images, resulting in query image-conditioned visual prompt features. These features are then grouped by class to form conditioned rich class-aware vision features. Then, we integrate the conditioned class-aware vision feature and language features using the module Class-aware Vision and Language Interleave (CVLI), which is added at the beginning of high layers of the language encoder. This interleave module consists of two blocks, each containing a cross-modality multi-head attention module (X-MHA) layer, correlating the hidden states of text with vision information, followed by an extra dense feed-forward (FF) layer. A tanh-gating mechanism [11] is used to improve training stability as we insert the module into a frozen pre-trained model and filter the low-quality visual features. To align the visual prompts and the predicted objects, a contrastive classification loss is used. To learn more knowledge from the conditioned class-aware vision features and avoid learning inertia problem, we use a random language token mask train-

ing strategy [47,42], which randomly masks the text tokens by a certain ratio to allow the corresponding vision queries to make predictions independently.

This study provides the following contributions:

- Building upon a frozen pre-trained vision-language (VL) detection model, we implement an open-set detection model that supports both textual and visual prompt queries through two cross-attention-based modules adaptation: a Vision Perceiver module for integrating query images with visual prompts, and a class-aware VL interleave module for aligning visual and textual features.
- In the CVLI module, we introduce a gating mechanism to filter out low-quality images and employ a gated loss function to further enhance model performance.
- Experiments demonstrate that our method is more transferable to downstream datasets. It outperforms GLIP-T/-L by +4.6%/2.7%, +5.7%/8.1%, and +0.0%/1.2% AP on 'LVIS minival' (MiniVal), 'LVIS val v1.0' (LVISv1), and COCO without model fine-tuning. It also improves few-shot transfer scenarios, and we evaluate the model on a CoalMine dataset.

2. Related Works

2.1. Few-shot Object Detection

Few-Shot Object Detection (FSOD) method aims to train a generic detector to recognize novel object detection with only a few training examples. The detec-

tors are typically trained with abundant samples of base classes and fine-tuned on few-shot novel samples. There are two main streams of traditional image-only-based FSOD methods: meta-learning-based and transfer-learning-based methods. The meta-learning-based methods simulate few-shot scenarios, containing two branches to process query and support images, respectively, and forming an episode format. FSRW [13] uses a meta-features learner to extract generalizable meta-features to detect novel objects. It uses a reweighting module to adjust meta-features and highlight more important and relevant ones to detect the target object. Meta R-CNN [43] performs meta-learning over the region of interest (RoI) features instead of a full image feature. Moreover, class-attentive vectors are generated using support images and are used to perform a channel-wise soft-attention on each RoI feature of query images, facilitating the predictor heads to detect or segment related class objects. FCT [9] also introduces a two-branch approach. The first approach proposes the vision transformer-based FSOD model, a fully cross-transformer for both the feature backbone and the detection head, encouraging multi-level interactions between the query and support. ICPE [25] generates specific and representative prototypes for each query image. Query perceptual features are fed into a prototype dynamic aggregation module, consisting of intra- and inter-image dynamic aggregation mechanisms to consolidate salient information of prototypes.

Transfer-learning-based FSOD methods' popularity stems from their superior performance, their modules' simplicity, and removing episode mechanisms that are used in meta-learning methods. For example, TFA [38] utilizes a cosine similarity-based classifier and only fine-tunes the last layer with novel examples, achieving comparable results with other complex methods. DeFRCN [29] employs gradient decoupled layers into Faster R-CNN for multi-stage decoupling, using an offline prototypical calibration block to refine the classification results. MFDC [40] designs a unified distillation framework based on a memory bank to distill three types (recognition-related semantic, localization-related semantic, and distribution) of class-agnostic commonalities between base and novel classes explicitly. With the development of transformer technology in vision, transformer learning-based methods in FSOD have attracted

significant research interest, especially with the rise of the vision-language large model.

We introduce the branch of visual prompts and use cross-attention to fuse the query image features, similar to the first method above. However, our model is based on visual and language modalities using visual prompts/support images to augment query images and generate class-aware vision features instead of using features for classification.

2.2. Open-vocabulary Object Detection

Open-vocabulary object detection, as a novel formulation of the object detection problem, was first proposed in OVR-CNN [48]. It attempts to cover an unbounded vocabulary of object concepts with the help of a large number of image-caption pairs so that the detector is no longer limited to a few categories with labeled data, leading to more generalized object detection that can recognize novel object categories. Subsequently, there was an increasing amount of work on OVD, and one pipeline is based on large-scale external image datasets [12, 19, 23, 51]. MDETR [12] trains an end-to-end model on existing multimodal datasets that explicitly align phrases in text and objects in images. GLIP [19] formulates object detection as a grounding problem. It leverages additional grounding data to learn aligned semantics at the phrase and region level, achieving even better performance on fully supervised detection benchmarks without fine-tuning. GroundingDINO [23] integrates Transformer-based DINO [49] with grounded pre-training and designs three feature fusion approaches in the neck, query initialization, and head phrases to help the model achieve better performance on existing benchmarks effectively. Our work is based on the GLIP model, an extension of these pipelines.

Another pipeline is to use the knowledge of large-scale pre-trained vision-language models, such as CLIP, to improve performance for recognizing novel objects, alongside the external data [7, 27, 37, 41, 47, 52]. ViLD [7] uses pre-trained CLIP (teacher) to distill knowledge into a two-stage (R-CNN-like) object detector (student). Moreover, region embeddings of detected boxes from the student detector are aligned with the text and image embeddings inferred by the teacher. OV-DETR [47] uses image and text embeddings encoded from a CLIP model as queries to decode the category-specified boxes in the DETR

framework [2], which can detect any object, given its class name or an exemplar image. RegionCLIP [52] proposes a region-based vision-language pre-training method, which learns to match image regions as well as their descriptions that learn the visual representation of regions from ‘pseudo’ region-text pairs from the CLIP model. HierKD [27] and OADP [37] employ knowledge distillation from CLIP. HierKD [27] explores a global-level knowledge distillation and combines it with the common instance-level knowledge distillation to learn the knowledge of seen and unseen categories simultaneously. OADP [37] employs object-level distillation and global and block distillation methods.

In the first pipeline of OVD, models are pre-trained on large-scale datasets from scratch, and the focus is on designing models for detection. Moreover, the second pipeline is based on CLIP, a model aimed at image classification, and it focuses on utilizing the knowledge of CLIP. Compared to the second pipeline, the first pipeline evaluates the model on public datasets, such as LVIS [8], COCO [22], and VOC [6]. It also assessed the model on the datasets in the wild, such as ODinW (Object Detection in the Wild). We follow the first, more practical pipeline in the real world and verify the model’s ability to adapt to novel classes at low shots.

3. Method

This section details our method, starting with the OVD settings and GLIP review. Then, we introduce an overview of our model architecture, the newly added modules, and their utilities. Finally, we illustrate the pre-training and fine-tuning pipelines.

3.1. Preliminaries

In OVD, an object detector is typically trained on a base dataset D_{train} , which contains exhaustively annotated bounding box labels of base categories C_b . During inference, to detect the categories of both base categories C_b and novel categories C_n , i.e., $C_{\text{test}}=C_b \cup C_n$. In academic research, evaluating a model’s performance usually requires there to be no overlap between the base and novel categories, and only the performance on novel categories is evaluated. However, in the real world, it is common to trans-

fer to downstream datasets that include both base and novel categories, such as our DsLMF+ datasets.

The parallel vision and language formulation are the mainstream architecture of existing VL detection foundation models [3, 19, 23, 32]. It unifies detection and grounding by reformulating object detection as phrase grounding, which attempts to localize objects and align them with semantic concepts.

Training data in OVD is usually in the form of image-text pairs (I, T) . We concatenate all category names as text input for the object detection task, which contains a set of unique classes. Then, we extract multi-scale image features with an image encoder Enc_I , such as SwinTransformer [24] and text features with a text encoder Enc_L , such as BERT [4]:

$$O = \text{Enc}_I(I), T = \text{Enc}_L(T), \quad (1)$$

where $T \in R^{M \times d}$ denotes the contextual word/token features from the language encoder. M is the number of (sub)-word tokens that are always larger than the number of phrases N_c (corresponding to classes numbers) in the text prompt.

Next, a deep fusion between image features O and text T is performed in the last few encode layers. This process uses the DyHead [3] as the image encoder and BERT [4] as the text encoder. It involves cross-modality fusion through a multi-head attention module (X-MHA), which outputs high-quality language-aware visual representations:

$$O_{i2i}^i, T_{i2i}^i = \text{X-MHA}(O^i, T^i), \quad (2)$$

where $i \in \{0, 1, \dots, L-1\}$

where L is the number of DyHeadModules, $L=6$ for GLIP-T and $L=8$ for GLIP-L.

Single modality fusion and update as follows:

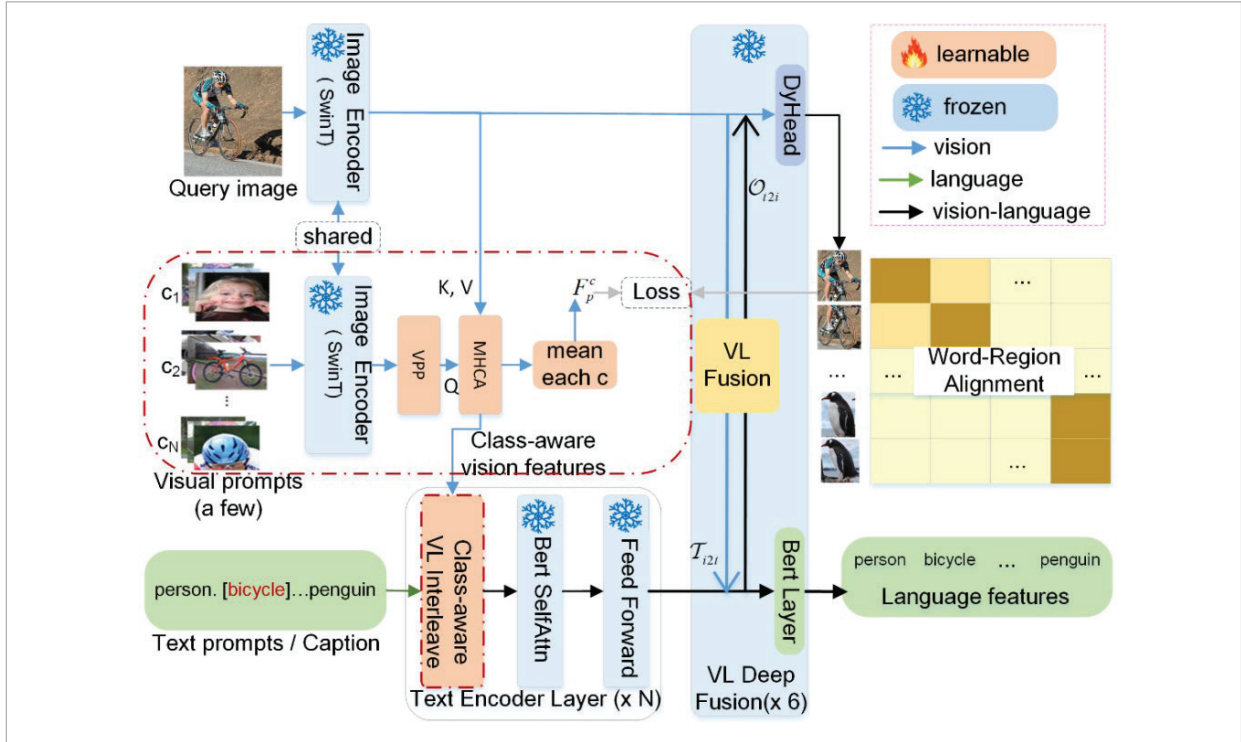
$$O^{i+1} = \text{DyHeadConv}_i(O^i + O_{i2i}^i) \quad (3)$$

$$T^{i+1} = \text{BERTLayer}_i(T^i + T_{i2i}^i). \quad (4)$$

Finally, normal box regression is applied for localization, and a vision-language dot product layer replaces the traditional linear classification layer to calculate the alignment scores between image regions and words in the prompt:

Figure 2

Architecture of our model. We add two learnable modules: The first is vision fusion between the query image features and the visual prompts features through MHCA. The second is class-aware VL interleave inserted in high-level text encoder layers to pre-align class-level vision features with class text embedding.



$$S_{ground} = O^L (T^L)^i. \quad (5)$$

The GLIP model has been trained on massive data, demonstrating strong zero-shot and few-shot transferability to various object-level recognition tasks. We take this model as our baseline.

3.2. Model Architecture

Considering the limited description of text-only, we take advantage of the visual prompt to enrich visual cues, enhance language descriptive capabilities, integrate the advantage of text prompt in generality, and enhance abstraction capabilities. The model architecture is shown in Figure2. Compared with the pre-trained model, the added components are indicated with a dash-dot line.

3.2.1. Cache the Visual Prompt Features

Visual prompt features are extracted and cached to save training time, as our model is based on frozen

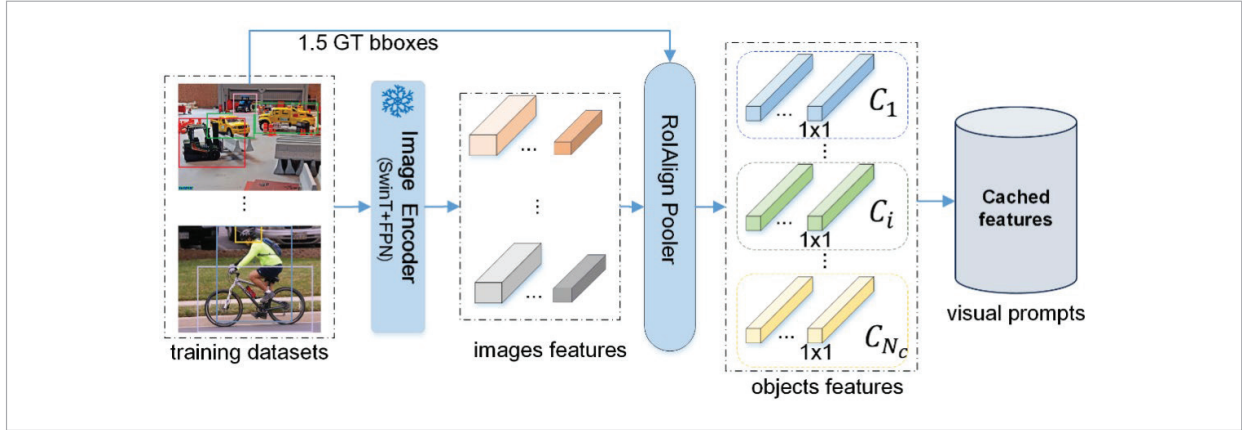
image encoder (SwinTransformer), which makes it feasible. We extract all features from training data and cache them before starting training.

Given a training dataset D_{train} that include categories C (pre-training: $C = C_b$; fine-tuning: $C = C_b + C_n$). Then, the instance annotations are region-text pairs, $O = \{b_i, t_i\}_{i=1}^N$ where t_i is the corresponding text for the region box b_i ; t_i can be the category name, noun phrases, or object descriptions. As a result, we adopt the category names as text prompts during pre-training and fine-tuning. The caching procedure is shown in Figure3.

Specifically, for an image, a ground truth (GT) box $b \in \mathbb{R}^4$ of category c_i is fed into the image encoder to obtain image features. Then, the box/object feature is aligned and fed into a RoI pooler [35] layer to generate object features $f^c \in \mathbb{R}^d$. We enlarge the GT box of 1.5 area to include more contextual information. Afterward, we aggregate these object features by category and construct a cached bank in a dictionary for-

Figure 3

Cache features of visual prompts. Extract image features with the shared frozen image encoder and pool them into $1 \times 1 \times d$, ($d=256$) shape. To reduce the computation time, we use offline computation and then caching for features.



mat: $B = \{ID_{c_i}, \{f_1, \dots, f_{N_o}\}\}, i \in \{1, \dots, N_c\}$, where the class index ID_{c_i} serves as the key, and all object features of class c_i are stored as values. N_c denotes the total number of categories and N_o represents the number of objects belonging to the category c_i .

During pre-training, as in GLIP [19], we limit the input length of BERT [4] to encode most 256 tokens. Based on this, we cannot fit all category names into one prompt when the dataset contains many categories, such as Object365 and LVIS. We also split the category names into multiple prompts such as GLIP. Specifically, in a minibatch, the GT categories of query images are noted as positive categories C_{pos} , and their corresponding total tokens lengths are noted as N_{pos} . We randomly selected $N_{neg} = 85$ or a randomly generated number from 1-85 among the remaining categories $C - C_{pos}$ with probability 0.5. Then, we append the tokens of these negative categories until the token length of all positive and negative reaches the maximum length $256 - N_{special} - N_{pos}$, where $N_{special}$ contains the start, end, and separation tokens.

Finally, the selected negative categories C_{neg} with number $N'_{neg} \leq N_{neg}$, because some category names have multiple words or some single-word phrases are split into multiple (sub)-word tokens.

The selected categories of support images for the visual prompts branch are $C_{pos} + C_{neg}$. At the fine-tuning stage, we simply concatenate all category names with a separator of ' '. when the dataset contains fewer categories, such as VOC or ODinW.

3.2.2. Visual Prompt Processing

Different visual prompts exert varying impacts on the prediction outcomes. Moreover, visual prompts of low quality can adversely affect the accuracy of predictions. Therefore, we employ self-attention mechanisms to capture the inter-feature correlations within the visual prompt features F_p , adjust the weights of the feature representations accordingly, and obtain the reweighted visual prompt features \hat{F}_p . The process is shown in Equation (6):

$$\begin{aligned} Q &= W_q F_p, K = W_k F_p, V = W_v F_p \\ A &= \text{soft} \max(QK^T / \sqrt{d / N_h}) V, \\ \hat{F}_p &= F_p + FF(AF_p) \end{aligned} \quad (6)$$

where d is the feature dimension and N_h is the number of heads. It is important to note that the above process is performed within features of the same class.

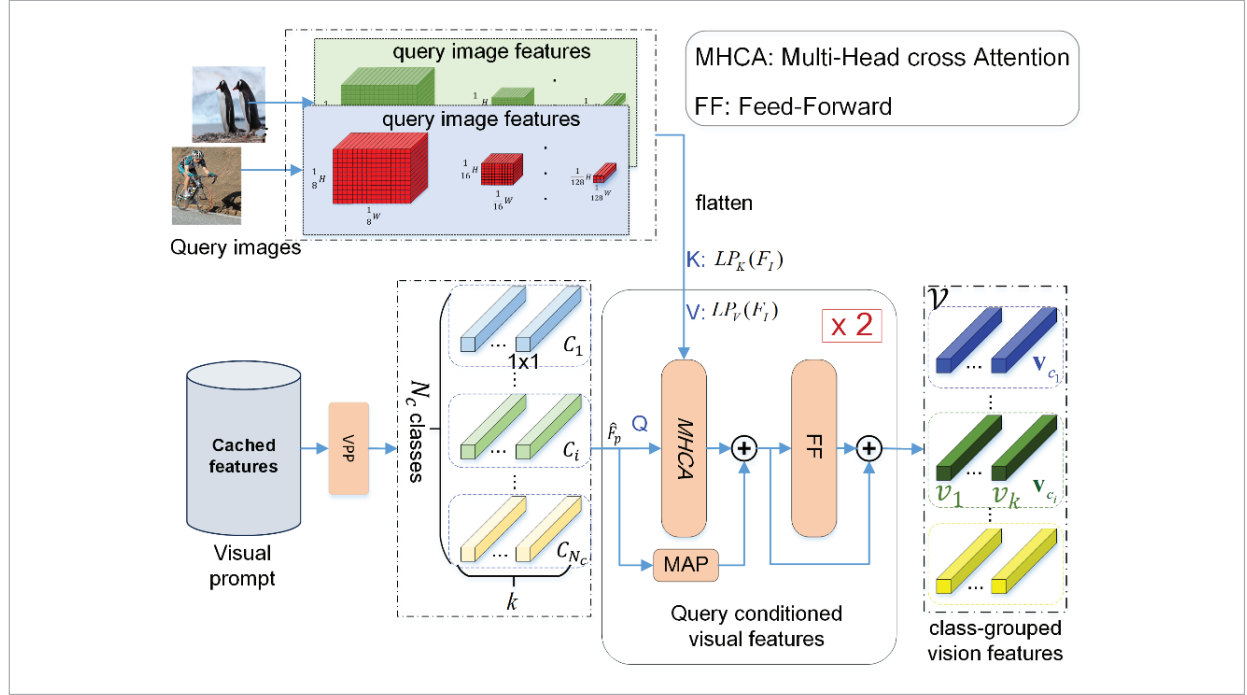
During training, we randomly sample a small number (e.g., 5) of cropped objects as image prompts for each class, and update them in each iteration. The random sampling allows the model to adapt to cross-domain image prompts, thereby enhancing the model's generalization ability.

3.2.3. Vision Perceiver Module

In the Visual Perceiver module, the image features are fused to the weighted visual prompt features using a masked cross attention to get the query conditioned visual prompt features.

Figure 4

Fuse the query image and support image to generate query image features conditioned class-grouped vision features.



As shown in Figure 4, the module consists of two blocks containing a multi-head cross-attention layer, which attends to the query inputs, and an additional FF layer. We take the visual prompt features as Query denoted as $\hat{F}_p \in \mathbb{R}^{B \times M_q \times d}$. Moreover, the key and value are from the query image features, denoted as $LP_K(F_I) \in \mathbb{R}^{B \times M_k \times d}$ and $LP_V(F_I) \in \mathbb{R}^{B \times M_v \times d}$, where B is the batch size, M_q is the number of visual prompt features, which could be $k \times N_c$ and M_v is the flattened input image features after the FPN [21]. The process of the block is as follows:

$$\begin{aligned} MHCA &= MHA(\hat{F}_p, LP_K(F_I), LP_V(F_I)) \\ CA &= MHCA + MAP(\hat{F}_p) \\ V &= FF(CA) + CA \end{aligned} \quad (7)$$

Layer normalization is applied to the keys, values, and queries that are the input to the attention and FF input, and a linear transform is applied to them before cross-attention. The final fused vision is the input image coupled vision features, having both input image vision cues and class level vision cues, represented by class as follows:

$$V = \mathbf{v}_i : \{v_1, \dots, v_k\}, i \in \{1, \dots, N_c\}. \quad (8)$$

The sequence length of class-wise vision features ($k \times N_c$) is much lower than that of the input image features $\sum_i H_i \times W_i$.

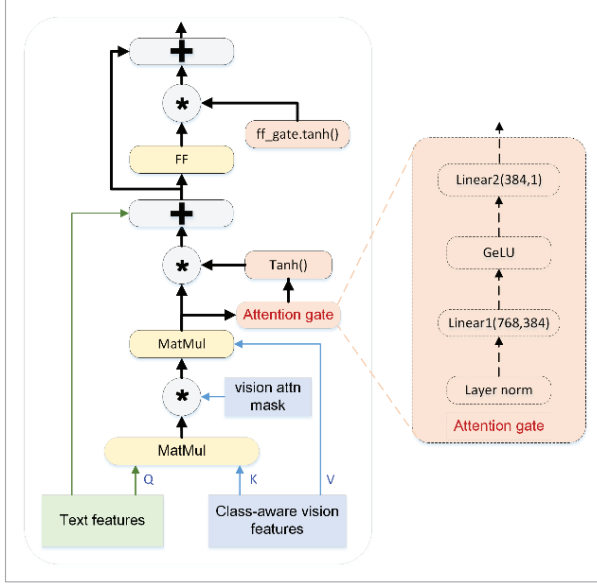
By averaging the k visual features, the class-level visual feature for each class is obtained. The class-level visual features for all classes are:

$$\bar{V} = \{\bar{v}_i\}, i \in \{1, \dots, N_c\} \quad (9)$$

3.2.4. Cross Modality Class-aware VL Interleave

We insert gated cross-attention blocks at the beginning of high language encode layers, trained from scratch to improve the expressiveness of VLM and make it sufficient to condition on visual inputs, inspired by Flamingo [1]. Those blocks consist of a cross-attention layer that attends the visual inputs with specific cross-attention masks, followed by an additional FF layer. The process is shown in Figure 5. The text encoder extracts text features for each category, $T = Enc_L(T) = \{t_c\}_{i=1}^{N_c}$. Using the class-aware vi-

Figure 5
Class-aware vision-language interleave.



sion features generated from Equation (8), the cross modalities multi-head attention (X-MHA) between vision and language is formulated as follows:

$$\tilde{v}_{c_i} = \mathbf{X}\text{-MHA}(t_{c_i}, \mathbf{v}_{c_i}), \quad i \in \{1, \dots, N_c\}, \quad (10)$$

where t_{c_i} is from the language modality and acts as the Query; \mathbf{v}_{c_i} is from the vision modality and acts as Key and Value.

A tanh-gating mechanism [11] is used to ensure that the model output is consistent with the pre-trained language model and yields the same results as the original language model. This process multiplies the output of the cross-attention layer through right $\tanh(\beta)$ before adding it to the input representation from the residual connection, where β is a layer-specific learnable scalar initialized at 0. The gating mechanism maintains the training stability and improves the final performance. The interleaved category tokens are gated so that the frozen text encoder remains intact at initialization, maintaining stability and improving generalization performance.

$$t'_{c_i} = t_{c_i} + \tanh((MLP(\tilde{v}_{c_i})) * \tilde{v}_{c_i}), \quad (11)$$

where the gating value $\tanh((MLP(\tilde{v}_{c_i})) * \tilde{v}_{c_i})$ is dynamically adjusted according to the quality of visual cues from the fused vision features, which is evaluated through a

three-layer perceptron (MLP). This framework gradually reduces the feature dimension to a layer-specific learnable scalar initialized at 0. Then, this learned scalar multiplies the multimodal feature \tilde{v}_{c_i} before adding it to the initial text token t_{c_i} from the residual connection. The final output after FF is processed by:

$$\tilde{t}_{c_i} = t'_{c_i} + FF(t'_{c_i}) * \tanh(\text{ff_gate}) \quad (12)$$

where ff_gate is a learnable parameter.

3.2.5. Training, Fine-tuning, and Inference

At the training stage, we train only the newly added modules while keeping the remaining parameters frozen. The training dataset used is Object365. The cache bank stores object-level image features for each class. A vision-conditioned masked language prediction strategy, as introduced in MQ-Det [42], is employed to facilitate training. This approach enables the model to learn from vision as the new module is integrated into the frozen language branch of the foundation model. It addresses the learning inertia issue caused by the frozen detector, which prevents the model from aligning regions with text features without visual cues. Specifically, the strategy randomly masks text tokens at a certain ratio, allowing corresponding vision queries to make independent predictions, thereby incorporating sufficient visual information.

The total loss function is as follows:

$$L = L_{loc} + L_{dot} + L_{cls} + L_{gate}, \quad (13)$$

where L_{loc} is the localization loss, which includes a GIoU loss [33] and a centerness loss [35].

L_{dot} plays a critical role in the classification logits, a focal binary sigmoid loss.

$L_{dot} = \text{TokenSigmoidFocalLoss}(S_{ground}, T')$, where the logits S_{ground} are the alignment scores in Equation (5), T' is the expanded target matrix.

Additionally, to achieve object-level class feature alignment between the visual prompts and the predicted objects, a classification loss function is introduced, which uses a contrastive loss to accomplish this:

$$L_{cls} = -\frac{1}{N_o} \sum_{i=1}^{N_o} \log \frac{\exp(o_i^T \bar{v}_{c_i} / \tau)}{\sum_{j=1}^{N_c} \exp(o_i^T \bar{v}_{c_j} / \tau)}, \quad (14)$$

where N_o denotes the number of objects.

L_{gate} is gate loss in the CVLI module, written as follows:

$$\lambda * \frac{1}{N_g} \sum_{i=1}^{N_g} (1 - |g_i|), \quad (15)$$

where λ is a hyperparameter; N_g is the number of CVLI inserted into the text encoder layers.

We freeze our pre-trained model and perform prompt fine-tuning at the few-shot fine-tuning stage, processing the data that are available for few-shot training. Due to the limited training data (k-shot for each class), the cache bank is constructed using all objects in the training set, and random select k-shot at each iteration.

During the inference phase, when evaluating the performance of the trained and fine-tuned models, if operating under the fine-tune-free setting, only one visual prompt is used per class. In contrast, under the k-shot fine-tuning setting, the same visual prompts used during fine-tuning are employed for evaluation.

4. Experiments

4.1. Datasets and Evaluation Metrics

4.1.1. Objects365 Dataset

Objects365 dataset [34] is a large-scale object detection dataset with 365 object categories and over 600K training images. It is also widely used in the OVD models [19,23,45,46,51]. We use it to train our additional modules.

4.1.2. LVIS and COCO Benchmark

LVIS dataset [8], designed for researching Large Vocabulary Instance Segmentation, which involves collecting instance segmentation masks for 1203 entry-level object categories, is also widely used in OVD. Evaluation results on a dataset with a large set of diverse object categories would be more representative in terms of demonstrating generalization ability. The LVISv1 set contains 19,809 images, while MiniVal is introduced in MDETR [12] with 5000 images. The MS-COCO (COCO) [22] object detection dataset contains 80 common object categories. It is also used to verify the fine-tuning free transfer capabilities of the model.

4.1.3. ODinW Benchmark and CoalMine Underground Dataset

ODinW Benchmark [17] includes 13 object detection datasets, spanning fine-grained species detection, drone-view detection, and ego-centric detection. It is used to evaluate the model's transferability in diverse real-world tasks. Then, we evaluate the model's fine-tuning-free performance on 13 ODinW datasets. Moreover, we select Pscal VOC from these 13 ODinW datasets to ensure a fair comparison with other methods for fine-tuning analysis. Following the few-shot split of different seeds in GLIP, the datasets contain 13,690 training images of 20 categories.

We introduce another real-world underground mining dataset, reorganized based on DsLMF+ [44], to evaluate the model's performance in a world containing common and rare specialized categories. Specifically, we integrate images and resample images from all separate sub-datasets to construct a new dataset containing 4,824 training images and 1,342 validation images of 4 categories (coal miner, helmet, towline, and hydraulic support guard plate). This dataset is more challenging because of the following issues: 1) images have different domain attributes even under the same category, as shown in Figure 6, and also have issues such as blurriness, darkness, and exposure; 2) the small object issue; and 3) new terminologies. Therefore, we use these two datasets to demonstrate the few-shot transferability of our model.

Figure 6

Images comparison between pre-trained Objects365 dataset and downstream fine-tuning sub-DsLMF+ dataset.



4.1.4. Evaluation Metrics

We use the COCO AP (IoU = 0.50: 0.95) when conducting fine-tuning-free experiments. Moreover, we employ the mAP50 (IoU = 0.50) metrics for fine-tuning experiments to evaluate our algorithms, which can demonstrate the fluctuation among different seeds more clearly.

4.2. Implementation Details

We train the model on the Object365 training dataset during pre-training for only one epoch using 4 A40 GPUs, with a batch size of 16 and 8 for the -T and -L models respectively. We use a base learning rate of 1×10^{-5} for the language backbone and query branch, 5×10^{-3} for gated learning, and 1×10^{-4} for all other parameters. The learning rate decreases by 0.1 at 95% of the total training steps.

At the fine-tuning stage, where the pre-trained model is transferred to other datasets, we first conduct OVD under a fine-tuning-free setting on MiniVal and COCO, directly demonstrating the model's generalization. Then, we performed prompt tuning [16] concerning deployment efficiency, involving tuning the least parameters for the best performance. We train the model with a learning rate of 0.05, batch size of 4, and weight decay of 0.25. The visual prompts are extracted from the few/full-shot training set.

For the LVIS and Objects365 datasets, which have

many categories, visual prompts are selected through a two-step process: First, categories are selected as determined by $C_{pos} + C_{neg}$. Second, the exemplars are randomly selected from the cached bank based on these categories. The ratio of masked language prediction in the interleave module is set to 0.4.

4.3. Fine-tuning Free Transfer on COCO and LVIS

4.3.1. Experimental Results

We evaluate the model's transferability to common categories on the COCO val2017 dataset. Under the zero-shot domain transfer setting, we evaluate models modified from GLIP-T and GLIP-L. Results are presented in Table 1, column 4, where strong zero-shot performance is achieved.

Moreover, we evaluate our pre-trained model's ability to detect rare and diverse objects on LVIS at zero-shot settings. We report performance on MiniVal and the full validation set v1.0. Results are presented in zero-shot domain transfer to MiniVal and LVISv1 without model fine-tuning. Results in Table 1 demonstrate that our model has strong transferability to other datasets with less training data. Our model also has fewer training parameters because we only trained the newly added modules which is more efficient. Compared to GLIP-T, our model obtains +4.6% AP on MiniVal, and +5.7% AP on LVISv1.

Table 1

Fine-tuning free performance on MiniVal, LVISv1, and COCO val2017. AP_r / AP_c / AP_f indicate the AP values for rare, common, frequent categories, respectively.

Model	Backbone	Pre-train data	COCO	MiniVal (%)				LVISv1 (%)			
			AP (%)	AP _r	AP _c	AP _f	AP	AP _r	AP _c	AP _f	AP
MDETR	RN101	GoldG,RefC	-	20.9	24.9	24.3	24.2	7.4	22.7	25	22.5
MaskRCNN	RN101	-	-	26.3	34	33.9	33.3	-	-	-	-
SuperRFS	RN50	-	-	-	-	-	-	12.3	24.3	32.4	25.4
DyHead-T	Swin-T	O365	43.6	-	-	-	-	-	-	-	-
GLIP-T (C)	Swin-T	O365,GoldG	46.7	17.7	19.5	31.0	24.9	7.5	11.6	26.1	16.5
GLIP-T	Swin-T	O365,GoldG,CC4M	46.5	20.8	21.4	31.0	26.0	10.1	12.5	25.5	17.2
G-DINO-T	Swin-T	O365,GoldG,Cap4M	48.4	18.1	23.3	32.7	27.4	-	-	-	-
Ours-T	Swin-T	O365	46.5	23.9	27.0	35.0	30.6	16.3	18.6	30.6	22.9
GLIP-L	Swin-L	FourODs,GoldG,Cap24M	49.8	28.2	34.3	41.5	37.3	17.1	23.3	35.4	26.9
Ours-L	Swin-L	O365	51.0	32.0	37.2	44.1	40.0	27.8	31.9	41.6	35.0

It should be noted, however, that there are some categories of LVIS that overlap with Object365 and contain all the categories of COCO. Our model is trained based on frozen GLIP-T/-L, and although it is only trained on Object365, the model still preserves its generalizability and achieves a better performance.

4.3.2. Qualitative Analysis

Our model is only fine-tuned on Objects365.

Table 2

Dataset size and parameters comparison.

Model	dataset size(M)	parameters(M)	
		Total	Trainable
GLIP-(T/L)	5.5/27.5	231.8/430.4	231.8/429.2
Ours-(T/L)	0.7/0.7	277.4/476.1	45.7/45.7

Table 3

Fine-tuning free performance on 13 datasets of ODinW.

Model	Backbone	Pre-train data	ODinW13 mAP (%)
MDETR	ENB5	GoldG,RefC	25.1
OWL-ViT	ViT L/14(CLIP)	O365,VG	40.9
DetCLIP-T	Swin-T	O365,GoldG,YFCC1M	43.3
OmDet	ConvNeXt-B	COCO,O365,LVIS,PhraseCut	43.6
GLIP-T	Swin-T	O365,GoldG,CC4M	41.9
G-DINO-T	Swin-T	O365,GoldG,Cap4M	49.8
GLIP-L	Swin-L	FourODs,GoldG,Cap24M	51.0
Ours-T	Swin-T	O365	45.1
Ours-L	Swin-L	O365	54.4

4.4.2. Fine-tuning on ODinW, VOC and CoalMine Datasets

We fine-tune our pre-trained model to evaluate its transferability to diverse real-world tasks. We select Pascal VOC to represent of ODinW because public baselines have been established on this dataset. We experiment with different amounts of task-specific annotated data, from zero-shot to few-shot and to all of the data in the training set. We sample the prompt image features from training data and perform prompt fine-tuning. Similar to the GLIP, we monitor the performance on validation and decay the learn-

ing rate by 0.1 when reaching the validation performance plateaus. Results are shown in Table 4. Compared to GLIP-T and GLIP-L, our models exhibited improved performances except for the 3-shot of the large (-L) model. Some commonalities can also be found from the Table: performance under 1-shot and 3-shot settings are worse than zero-shot on GILP-T and Ours-T, indicating that when the sample size is extremely small, overfitting can easily lead to performance degradation. Moreover, -L models have at least 5.0 and 5.2 points improvement over -T models, verifying that large models are more powerful.

4.4. Object Detection in the Wild

4.4.1. Fine-tuning Free Results on 13 ODinW Detection Datasets

We further transfer our models to ODinW detection datasets by fine-tuning free mode to investigate the generalization ability. The Average APs of 13 datasets are shown in Table 3. Compared to the GLIP (-T/-L) baseline, our method shows performance improvements of 45.1 vs. 41.9 and 54.4 vs. 51.0, respectively.

Table 4

Results on Pascal VOC, under the settings of fine-tuning free eval (zero-shot), few-shot and full data prompt tuning (PT).

VOC (AP50)	Zero-shot (%)	Few-shot (%)				Fulldata (%)
		1-shot	3-shot	5-shot	10-shot	
GLIP-T	70.2	66.5	70.0	71.1	73.9	82.5
Ours-T	72.3	66.9	71.3	72.6	75.3	83.4
GLIP-L	74.6	76.0	79.3	79.7	80.7	86.5
Ours-L	76.7	77.5	78.8	80.8	81.9	87.1

We then transfer our model to a real-world coal mining dataset, DsLMF+. This dataset contains the categories of person (coal miner) and helmet, which appear in the base classes, and the categories of towline and hydraulic support guard plate, which are in the novel (unseen) classes. Similarly to the VOC dataset, we conduct experiments on the settings of zero-shot, few-shot, and full data. Results are shown in Table 5.

Table 5

Results on CoalMine, under the fine-tuning free eval (zero-shot) settings, few-shot and full data prompt fine-tuning. LP: Linear Probing.

CoalMine (AP50)	zero-shot (%)	Few-shot (%)				full-data (%)
		1-shot	3-shot	5-shot	10-shot	
GLIP-T(LP)	29.9	29.7	30	30.1	33.9	64.6
GLIP-T	29.9	46.8	55.3	56.2	70.6	85.3
Ours-T	35.1	41.6	58.6	59.6	70.7	88.9
GLIP-L	28.1	43.2	60.8	59.4	69.7	86.0
Ours-L	37.4	48.2	65.7	62.0	74.0	90.4

From Tables 4-5, fine-tuning may lead to performance degradation in extreme low-shot cases, overfitting low shots and undermining the model's ability. Moreover, more training data available tends to get better performance. Finally, large model tends to have better performance.

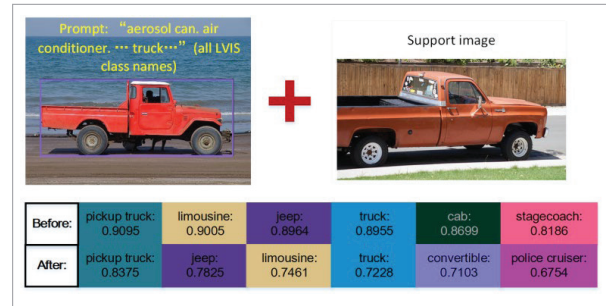
5. Ablation Studies

5.1. Importance of Visual Prompt

We perform experiments with and without visual prompts to analyze their impact on performance and

Figure 7

Comparing results with and without visual prompts on LVIS under settings of fine-tuning free.



the relationships between different visual prompts and query images.

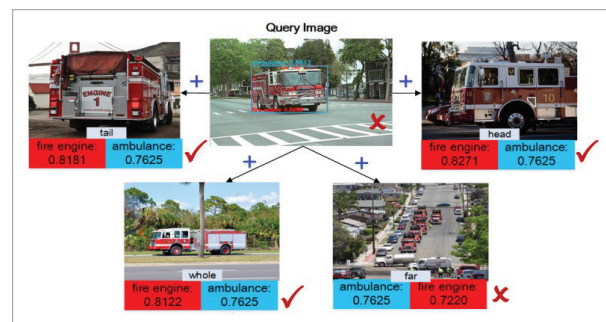
In Figure 7, before using the supported image, 'pickup truck' and 'limousine' had the same accuracy; 'jeep' and 'truck' had the same accuracy when only one decimal place was kept. After using the visual prompts, the accuracy of the target is clearly distinguished. It can be observed that the confidence predicted by GLIP is generally overestimated, and it fails to distinguish between classes when their similarity is high. In contrast, our method effectively amplifies.

We analyze the effect of different visual prompts on the query image of the same class in Figure 8.

We evaluate the detection results with the text prompt of all LVIS class names under the LVIS fine-tuning free settings. We can see that the result is wrong if we evaluate the query image with zero-shot; the results in this case are 0.86 accuracy for 'ambulance' compared to 0.82 accuracy for 'fire engine'. With the visual prompts of 'head', 'tail' and 'whole', we could obtain

Figure 8

Analyze the effect of different angles of the visual prompts on prediction results. Text prompt: concatenation of all LVIS class names.



the correct results. However, the ‘far’ visual prompt still gives the wrong result, because of the ambiguity caused by small targets at a distance. The cosine similarities between the query image and different visual prompts are shown in Table 6.

Table 6

Cosine similarities between the query image and different visual prompts.

Cosine Similarity	Support Images			
	Head	Tail	Far	Whole
Query Image	0.865	0.859	0.388	0.846

5.2. Analysis of the Effect of Loss Function

We analyze the impact of the newly added loss functions on model performance in Table 7, using the -T model on the Minival dataset. The results verify the effectiveness of the classification loss function, and the model performance is optimal when both are used in combination.

Table 7

Analysis of the impact of classification loss and gate loss on performance.

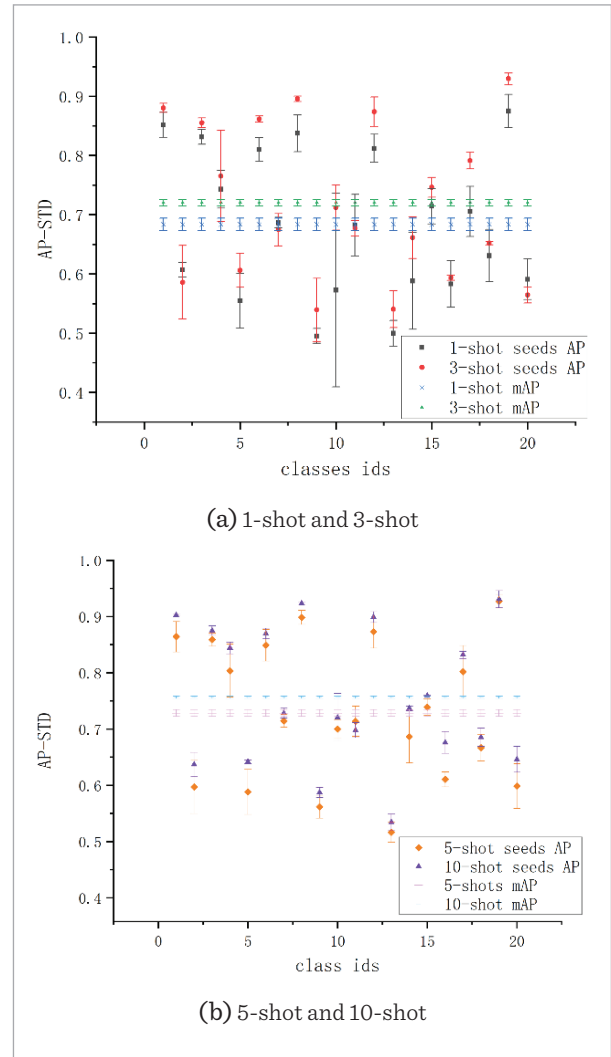
L_{gate}	L_{cls}	Minival (%)			
		AP_r	AP_c	AP_f	AP
-	-	20.8	21.4	31.0	26.0
✓	-	23.2	26.1	33.8	28.3
✓	✓	23.9	27.0	35.0	30.6

5.3. Performance Steadiness Analysis

Following GLIP [19], Grounding DINO [23], we use seed 3, seed 30, and seed 300, which was provided on ODinW datasets. The average precision (AP) of different categories and standard deviation (STD) among different seeds and show the mean AP (mAP) of all categories for different shots. The results are shown in Figure 9. We use AP50 (‘metric (AP50) [IoU=0.50|area=all | maxDets=100]’) for evaluation. In the figure, each point represents the average precision for a specific class across three random seeds, and the length of the error bar ($2 \times STD$) indicates the range of performance variation. By comparing Figure 9(a) with Figure 9(b), we observe that the AP values gradually increase, while the corresponding STD values tend to decrease as the number of shots increases.

Figure 9

AP and STD to represent the accuracy and its corresponding uncertainty.



We sample different training images from CoalMine and constructed three few-shot training datasets, namely seed3, seed30, and seed300. We incrementally expand the samples, e.g., 3-shot contains all 1-shot samples, 5-shot contains all 3-shot samples, and so forth, to compare the model’s performance under different shots more clearly. We perform experiments on fine-tuning free evaluation and prompt tuning on the settings of different shots with different seeds. Results are shown in Table 8 with the metric of AP50. The results show that the model exhibits poor performance on new cat-

egories when directly evaluated without fine-tuning, especially on the 'guard plate', which exhibits greater diversity due to different perspectives of the samples as shown in Figure 6. Compared to zero-shot inference, 1-shot fine-tuning degrades the model's performance on the person category. A similar result is found in CLIP, where zero-shot CLIP matches the average performance of a 4-shot

linear classifier. We found that the AP values on the 'guard plate' vary greatly when using different samples because of the visual variability in the images at different angles. For example, the plates entirely differ at 90 degrees and 0 degrees. In this case, finer-grained classification is needed to divide the images of different angles into groups to improve the performance.

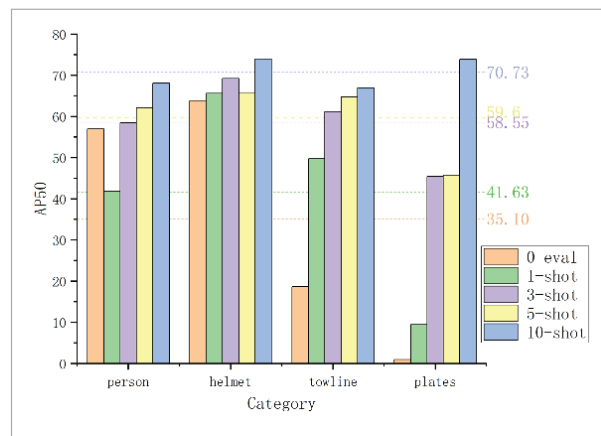
Table 8

Few-shot performance under different seeds on CoalMine dataset.

Few-Shots	seeds	Person (%)	Helmet (%)	Towline (%)	guard plate (%)	AVG (%)	Shots AVG (%)
0-eval	-	57.0	63.8	18.6	0.9	35.1	-
1-shot	seed3	41.7	65.6	49.7	9.5	41.6	47.2
	seed30	55.3	68.5	39.8	20.1	45.9	
	seed300	56.1	65.4	57.5	37.6	54.2	
3-shot	seed3	58.4	69.2	61.1	45.4	58.6	58.1
	seed30	62.9	67.8	68.1	25.2	56.0	
	seed300	50.4	65.8	62.4	59.9	59.6	
5-shot	seed3	62.1	65.8	64.7	45.8	59.6	61.7
	seed30	61.0	65.2	64.1	41.2	57.9	
	seed300	62.8	69.9	70.2	67.3	67.6	
10-shot	seed3	68.1	74.0	67.0	73.9	70.7	67.2
	seed30	67.6	65.3	71.1	42.5	61.6	
	seed300	68.0	76.9	72.1	60.0	69.3	

Figure 10

Performance of different shots on different categories on the CoalMine dataset at seed3. The horizontal line indicates the average accuracy across all classes under different shot settings.



We compare the performance of different categories among different shots under seed-3 settings for CoalMine. As shown in Figure 10, the model has poor performance on zero-shot inference settings when confronted with novel objects in a specific domain, with only 0.9% accuracy on the 'hydraulic support plate' class despite GLIP's strong general object detection capabilities. When fine-tuning the model with few-shot samples, the more samples that are made available will result in better performance. From the Figure 10, we can see that the performance on the new categories ('towline' and 'hydraulic support plate') improves rapidly from 0-shot to 10-shot.

5.4. Qualitative Visualization on CoalMine

We visually compare the detection results in the different training datasets as shown in Figure 11. From the 0-shot column, it can be seen that the novel class-

Figure 11

Visualization of different shot detection results on the CoalMine dataset.



es are not detected and the person in the third row is also missed. Fine-tuning with more shots yields better detection results.

5.5. Inference Time Comparisons

We compared inference time with baselines on COCO, CoalMine and VOC datasets under different shots settings. The results are shown in Table 9. Due to the addition of new modules to the model, the inference time has increased along with the improvement in performance.

5.6. VL Attention and Detection Generalizability Visualization

We extract the cross-attention weights between visual and language from the last VLFuse block of the VL deep

Table 9

The inference time comparisons. The unit is seconds per image.

Datasets	Shots	GLIP-T	Ours-T	GLIP-L	Ours-L
COCO	ft-free	0.28	0.25	0.30	0.44
CoalMine	ft-free	0.28	0.27	0.30	0.44
	1-shot	0.18	0.26	0.31	0.44
	10-shot	0.18	0.26	0.30	0.44
	all-data	0.18	0.28	0.30	0.45
VOC	ft-free	0.17	0.20	0.46	0.40
	1-shot	0.17	0.24	0.29	0.37
	10-shot	0.17	0.24	0.29	0.41
	all-data	0.17	0.27	0.29	0.41

fusion module and compute its gradCAM to illustrate the accuracy of VL alignment. We could find the corresponding visual gradCAM values of tokenized words through the visual mask. We average the corresponding gradCAM for a word containing multiple sub-word tokens for visualization. For example, 'deck chair' is split into 'deck' and 'chair.' 'Parasol' is split into 'para' and '\#\#sol' when tokenized with Bert encoder. Then, we average the gradCAM values of the corresponding 'deck' and 'chair' (or 'para' and '\#\#sol') for visualization. The results are shown in Figure 12.

One advantage of language-based detection models is that they can take advantage of the flexibility of language to detect only the targets of interest from the

user. From Figure 13, we can input different captions to detect different objects. In general, the datasets that are used to train the large model contain a very large number of categories, such as the image-text pairs used in GLIP-L, which retains 58.4M unique noun phrases even after filtering with high confidence (>0.5). From the middle one in Figure 13, 'sand' and 'sea' were also detected, which verifies that the model retained the generalizability of the pre-trained foundation model. In addition to the concat class names used as text input, we also show the results of inference when sentences are entered, as shown in the last image of Figure 13. Nouns are extracted using the NLTK library and then detected on test images.

Figure 12

Grad-CAM visualization on the Visual-Language alignment maps corresponding to input captions.

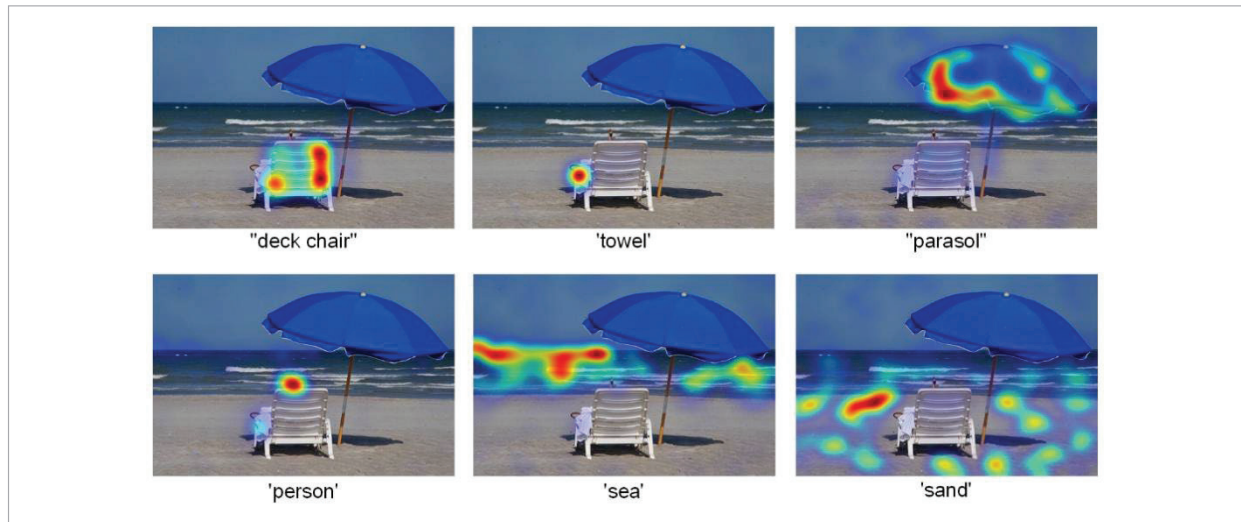
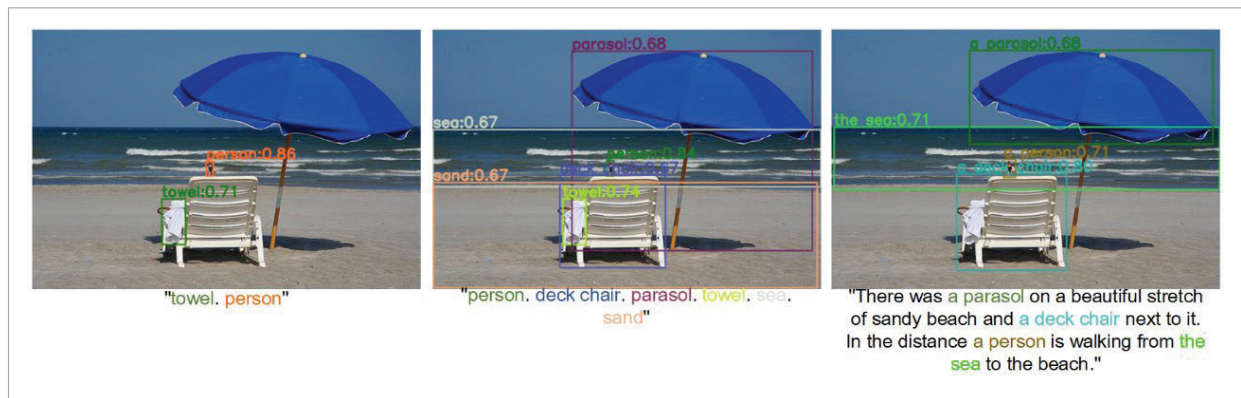


Figure 13

Detect the objects of interest flexibly and maintain the generalizability of the foundation model.



6. Discussion and Conclusion

Text-based queries are introduced in OVD methods, allowing users to flexibly input descriptive phrases to search for specific objects. Building upon a pre-trained OVD model, we incorporate a visual prompt branch to develop a model that supports both text and image queries for a given test image. This dual-modality model fully leverages available few-shot datasets and improves the performance of downstream few-shot object detection tasks. However, its performance under single-modality input is lower compared to when both modalities are used simultaneously. Therefore, maintaining high performance with dual-modality input while enhancing performance in single-modality scenarios is one of our future research goals. Additionally, due to the large number of parameters in OVD mod-

els, designing a lightweight downstream detector to improve inference speed is also an important direction for future work.

In this paper, we implement an efficient open-set detection model that can leverage both text and few of visual prompts as input. It employs the frozen pre-trained foundation model by plugging in few modules plug-and-play, preserving the model's generalizability and saving the cost. Visual prompts serve as visual cues, enhancing visual information and mitigating the limitations of text-only prompts, which can be dynamically adjusted according to downstream tasks, thereby improving the performance of the downstream tasks. The model leverages the clear visual presentation of specific details and the generalizability of text prompts. The experimental results show the effective transferability of foundation models to different downstream datasets.

Data Availability Statement

AVAILABILITY OF DATA	STATEMENT OF DATA AILABILITY
Objects365	https://www.objects365.org/overview.html
LVIS	https://www.lvisdataset.org/dataset
COCO	http://cocodataset.org/#download
ODinW	https://huggingface.co/GLIPModel/GLIP/tree/main/odinw_35
CoalMine	https://pan.baidu.com/s/1nsoEA1MsOxjtUrbVfc9PaQ , extraction code: 1111. The recollected and labeled CoalMine dataset is available from Qinghua Yang upon reasonable request.

Appendix A

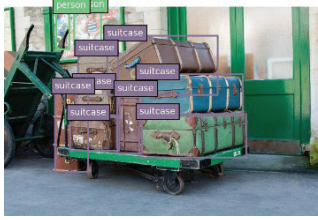
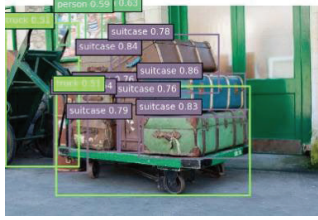

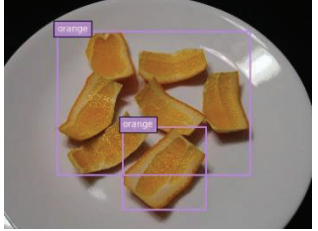
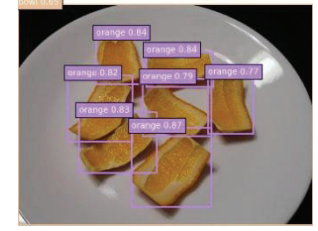
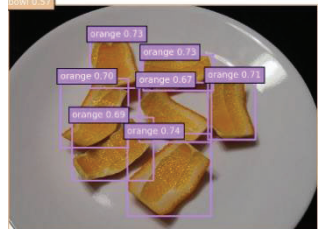
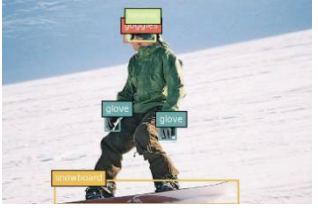
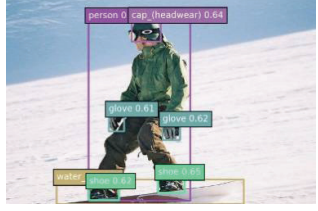



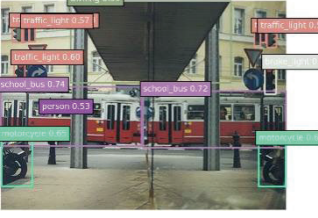



A1. Qualitative Visualization on COCO and LVIS

We visualize some detection results of model(-L) in Table A1. In the first row of COCO, our method can correct similar objects of misdetect problems in GLIP, such as the 'spoon' in the figure. In the second

row of COCO, both GLIP and our method can correct erroneous ground truth annotations. However, some of the classes in the validation set are unlabeled (e.g., 'person' and 'train') due to the unique setup of the LVIS dataset. However, the OVOD models can still detect these objects. Our model provides better performance results than the detection results on Mini-Val and LVISv1 with GLIP.

Table A1

Fine-tuning free detection results visualization on COCO Val, MiniVal and LIVSv1.

GT	GLIP	Ours
		
		
		
		
		





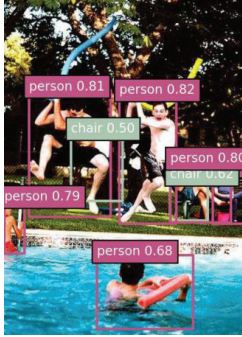
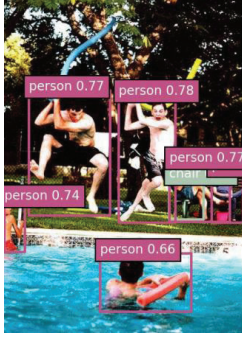
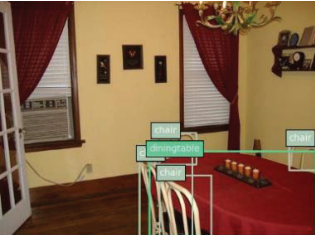

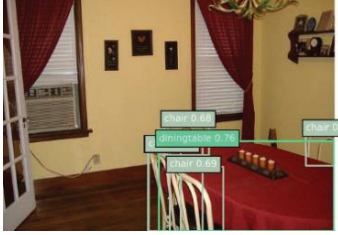
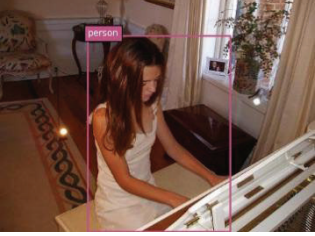
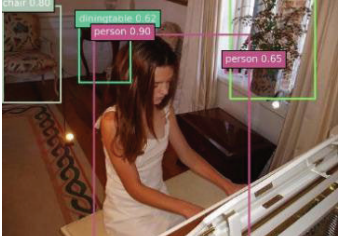
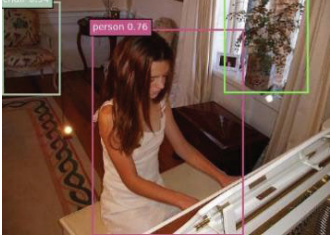
A2. Qualitative Visualization on VOC

We conduct model(-T) fine-tuning free inference and model(-L) full-data fine-tuning visualization on the VOC dataset and then the results, as shown in Table A2. Similar to Table A1, we still find that GLIP mis-detects similar objects, and our model

can alleviate this problem to obtain more accurate detection results. Both GLIP and our model can correct some errors in the ground truth, e.g., missing annotations of 'person' and 'chair' in the second row and missing 'chair' and 'potted plant' annotations in the fourth row.

Table A2

Comparison of VOC detection results with model(-T) without tuning and model(-L) with full data tuning.

GT	GLIP	Ours
		
		
		
		

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the cur-

rent study are available from the corresponding author on reasonable request.

Funding

This study was supported by the R&D Program of Beijing Municipal Education Commission (No. KM202211417005) and Academic Research Projects of Beijing Union University (No. ZK90202106).

References

1. Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z. T., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., Simonyan, K. Flamingo: A Visual Language Model for Few-Shot Learning. *Advances in Neural Information Processing Systems* 35 (NeurIPS 2022), 2022. [Online]. Available: <Go to ISI>://WOS:001215469503020.
2. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S. End-to-end Object Detection with Transformers. In *European Conference on Computer Vision*, 2020: Springer, 213-229. https://doi.org/10.1007/978-3-030-58452-8_13
3. Dai, X. Y., Chen, Y. P., Xiao, B., Chen, D. D., Liu, M. C., Yuan, L., Zhang, L. Dynamic Head: Unifying Object Detection Heads with Attentions. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2021)*, 2021, 7369-7378. <https://doi.org/10.1109/CVPR46437.2021.00729>
4. Devlin, J., Chang, M. W., Lee, K., Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019*, 1, 2019, 4171-4186. [Online]. Available: <Go to ISI>://WOS:000900116904035.
5. Du, Y., Wei, F. Y., Zhang, Z. H., Shi, M. J., Gao, Y., Li, G. Q. Learning to Prompt for Open-Vocabulary Object Detection with Vision-Language Model. *Proceedings of CVPR IEEE*, 2022, 14064-14073. <https://doi.org/10.1109/CVPR52688.2022.01369>
6. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 2010, 88(2), 303-338. <https://doi.org/10.1007/s11263-009-0275-4>
7. Gu, X., Lin, T.-Y., Kuo, W., Cui, Y. Open-vocabulary Object Detection Via Vision and Language Knowledge Distillation. *arXiv preprint arXiv:2104.13921*, 2021.
8. Gupta, A., Dollár, P., Girshick, R. LVIS: A Dataset for Large Vocabulary Instance Segmentation. *CVPR 2019*, 5351-5359. <https://doi.org/10.1109/CVPR.2019.00550>
9. Han, G. X., Ma, J. W., Huang, S. Y., Chen, L., Chang, S. F. Few-Shot Object Detection with Fully Cross-Transformer. *CVPR 2022*, 5311-5320. <https://doi.org/10.1109/CVPR52688.2022.00525>
10. He, K., Zhang, X., Ren, S., Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
11. Hochreiter, S., Schmidhuber, J. Long Short-term Memory. *Neural Computation*, 1997, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
12. Kamath, A., Singh, M., Lecun, Y., Synnaeve, G., Misra, I., Carion, N. MDETR - Modulated Detection for End-to-End Multi-Modal Understanding. *ICCV 2021*, 1760-1770. <https://doi.org/10.1109/ICCV48922.2021.00180>
13. Kang, B. Y., Liu, Z., Wang, X., Yu, F., Feng, J. S., Darrell, T. Few-shot Object Detection via Feature Reweighting. *ICCV 2019*, 8419-8428. <https://doi.org/10.1109/ICCV.2019.00851>
14. Khattak, M. U., Rasheed, H., Maaz, M., Khan, S., Khan, F. S. MaPLe: Multi-modal Prompt Learning. *CVPR 2023*, 19113-19122. <https://doi.org/10.1109/CVPR52729.2023.01832>
15. Krizhevsky, A., Sutskever, I., Hinton, G. E. Imagenet Classification with Deep Convolutional Neural Networks. *NeurIPS*, 2012, 25.
16. Lester, B., Al-Rfou, R., Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. *EMNLP 2021*, 2021, 3045-3059. [Online]. Available: <Go to ISI>://WOS:000855966303015 <https://doi.org/10.18653/v1/2021.emnlp-main.243>
17. Li, C. Y., Liu, H. T., Li, L. H., Zhang, P. C., Aneja, J., Yang, J. W., Jin, P., Hu, H. D., Liu, Z. C., Lee, Y. J., Gao, J. F. ELEVATER: A Benchmark and Toolkit for Evaluating Language-Augmented Visual Models. *NeurIPS 2022*, 2022. [Online]. Available: <Go to ISI>://WOS:001215469502049
18. Li, F., Jiang, Q., Zhang, H., Ren, T. H., Liu, S. L., Zou, X. Y., Xu, H. Z., Li, H. Y., Yang, J. W., Li, C. Y., Zhang, L., Gao, J. F. Visual In-Context Prompting. *CVPR 2024*, 12861-+. <https://doi.org/10.1109/CVPR52733.2024.01222>
19. Li, L. H., Zhang, P. C., Zhang, H. T., Yang, J. W., Li, C. Y., Zhong, Y. W., Wang, L. J., Yuan, L., Zhang, L., Hwang, J. N., Chang, K. W., Gao, J. F. Grounded Language-Image Pre-training. *CVPR 2022*, 10955-10965. <https://doi.org/10.1109/CVPR52688.2022.01069>
20. Li, Z., Li, C. F., Guan, T. X., Shang, S. P. Underwater Object Detection Based on Improved Transformer and

- Attentional Supervised Fusion. *Information Technology and Control*, 2023, 52(2), 397-415. <https://doi.org/10.5755/j01.itc.52.2.33214>
21. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. Feature Pyramid Networks for Object Detection. *CVPR 2017*, 2117-2125. <https://doi.org/10.1109/CVPR.2017.106>
 22. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L. Microsoft COCO: Common Objects in Context. *ECCV 2014*, PT V, 8693, 740-755. https://doi.org/10.1007/978-3-319-10602-1_48
 23. Liu, S. L., Zeng, Z. Y., Ren, T. H., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C. Y., Yang, J. W., Su, H., Zhu, J., Zhang, L. Grounding DINO: Marrying DINO with Grounded Pre-training for Open-Set Object Detection. *ECCV 2024*, Pt XLVII, 15105, 38-55. https://doi.org/10.1007/978-3-031-72970-6_3
 24. Liu, Z., Lin, Y. T., Cao, Y., Hu, H., Wei, Y. X., Zhang, Z., Lin, S., Guo, B. N. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. *ICCV 2021*, 9992-10002. <https://doi.org/10.1109/ICCV48922.2021.00986>
 25. Lu, X. N., Diao, W. H., Mao, Y. Q., Li, J. X., Wang, P. J., Sun, X., Fu, K. Breaking ImmuTable: Information-Coupled Prototype Elaboration for Few-Shot Object Detection. *AAAI 2023*, 37(2), 1844-1852. [Online]. <https://doi.org/10.1609/aaai.v37i2.25274>
 26. Luo, Z. Z., Jia, S. Y., Niu, H. J., Zhao, Y. F., Zeng, X. Y., Dong, G. H. Elderly Fall Detection Algorithm Based on Improved YOLOv5s. *Information Technology and Control*, 2024, 53(2). <https://doi.org/10.5755/j01.itc.53.2.36336>
 27. Mal, Z. Y., Luo, G., Gao, J., Li, L., Chen, Y. X., Wang, S. R., Zhang, C. X., Hu, W. M. Open-Vocabulary One-Stage Detection with Hierarchical Visual-Language Knowledge Distillation. *CVPR 2022*, 14054-14063. <https://doi.org/10.1109/CVPR52688.2022.01368>
 28. Ortner, T., Petschenig, H., Vasilopoulos, A., Renner, R., Brglez, S., Limbacher, T., Pinero, E., Linares-Barranco, A., Pantazi, A., Legenstein, R. Rapid Learning with Phase-change Memory-based In-memory Computing Through Learning-to-Learn. *Nature Communications*, 2025, 16(1). <https://doi.org/10.1038/s41467-025-56345-4>
 29. Qiao, L. M., Zhao, Y. X., Li, Z. Y., Qiu, X., Wu, J. N., Zhang, C. DeFRCN: Decoupled Faster R-CNN for Few-Shot Object Detection. *ICCV 2021*, 8661-8670. <https://doi.org/10.1109/ICCV48922.2021.00856>
 30. Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I. Learning Transferable Visual Models from Natural Language Supervision. *CVPR 2021*, 139. [Online]. Available: <Go to ISI>://WOS:000768182704084
 31. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *CVPR 2016*, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
 32. Ren, S. Q., He, K. M., Girshick, R., Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE TPAMI*, 2017, 39(6), 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
 33. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S. Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression. *CVPR 2019*, 658-666. <https://doi.org/10.1109/CVPR.2019.00075>
 34. Shao, S., Li, Z. M., Zhang, T. Y., Peng, C., Yu, G., Zhang, X. Y., Li, J., Sun, J. Objects365: A Large-scale, High-quality Dataset for Object Detection. *ICCV 2019*, 8429-8438. <https://doi.org/10.1109/ICCV.2019.00852>
 35. Tian, Z., Shen, C. H., Chen, H., He, T. FCOS: Fully Convolutional One-Stage Object Detection. *ICCV 2019*, 9626-9635. <https://doi.org/10.1109/ICCV.2019.00972>
 36. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. Attention Is All You Need. *NeurIPS*, 2017, 30.
 37. Wang, L. T., Liu, Y., Du, P. H., Ding, Z. H., Liao, Y., Qi, Q. S., Chen, B. L., Liu, S. Object-Aware Distillation Pyramid for Open-Vocabulary Object Detection. *CVPR 2023*, 11186-11196. <https://doi.org/10.1109/CVPR52729.2023.01076>
 38. Wang, X., Huang, T. E., Darrell, T., Gonzalez, J. E., Yu, F. Frustratingly Simple Few-shot Object Detection. *arXiv preprint arXiv:2003.06957*, 2020.
 39. Wang, Z. C., Yang, B., Yue, H. N., Ma, Z. H. Fine-Grained Prototypes Distillation for Few-Shot Object Detection. *AAAI 2024*, 38(6), 5859-5866. [Online]. <https://doi.org/10.1609/aaai.v38i6.28399>
 40. Wu, S., Pei, W. J., Mei, D. W., Chen, F. L., Tian, J. D., Lu, G. M. Multi-faceted Distillation of Base-Novel Commonality for Few-Shot Object Detection. *ECCV 2022*, PT IX, 13669, 578-594. https://doi.org/10.1007/978-3-031-20077-9_34
 41. Wu, X. S., Zhu, F., Zhao, R., Li, H. S. CORA: Adapting CLIP for Open-Vocabulary Detection with Region Prompting

- and Anchor Pre-Matching. CVPR 2023, 7031-7040. <https://doi.org/10.1109/CVPR52729.2023.00679>
42. Xu, Y., Zhang, M., Fu, C., Chen, P., Yang, X., Li, K., Xu, C. Multi-modal Queried Object Detection in the WILD. NeurIPS 2023, 36, 4452-4469.
43. Yan, X. P., Chen, Z. L., Xu, A. N., Wang, X. X., Liang, X. D., Lin, L. Meta R-CNN: Towards General Solver for Instance-level Low-Shot Learning. ICCV 2019, 9576-9585. <https://doi.org/10.1109/ICCV.2019.00967>
44. Yang, W. J., Zhang, X. H., Ma, B., Wang, Y. Q., Wu, Y. J., Yan, J. X., Liu, Y. W., Zhang, C., Wan, J. C., Wang, Y., Huang, M. Y., Li, Y. Y., Zhao, D. An Open Dataset for Intelligent Recognition and Classification of Abnormal Condition in Longwall Mining. Scientific Data, 2023, 10(1). <https://doi.org/10.1038/s41597-023-02322-9>
45. Yao, L. W., Han, J. H., Liang, X. D., Xu, D., Zhang, W., Li, Z. G., Xu, H. DetCLIPv2: Scalable Open-Vocabulary Object Detection Pre-training via Word-Region Alignment. CVPR 2023, 23497-23506. <https://doi.org/10.1109/CVPR52729.2023.02250>
46. Yuan, L., Chen, D., Chen, Y.-L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C. Florence: A New Foundation Model for Computer Vision. arXiv preprint arXiv:2111.11432, 2021.
47. Zang, Y. H., Li, W., Zhou, K. Y., Huang, C., Loy, C. C. Open-Vocabulary DETR with Conditional Matching. ECCV 2022, PT IX, 13669, 106-122. https://doi.org/10.1007/978-3-031-20077-9_7
48. Zareian, A., Dela Rosa, K., Hu, D. H., Chang, S. F. Open-Vocabulary Object Detection Using Captions. CVPR 2021, 14388-14397. <https://doi.org/10.1109/CVPR46437.2021.01416>
49. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., Shum, H.-Y. Dino: Detr with Improved Denoising Anchor Boxes for End-to-End Object Detection. arXiv preprint arXiv:2203.03605, 2022.
50. Zhang, X., Liu, Y., Wang, Y., Boularias, A. Detect Everything with Few Examples. arXiv preprint arXiv:2309.12969, 2023.
51. Zhao, T. C., Liu, P., Lee, K. OmDet: Large-scale Vision-language Multi-dataset Pre-training with Multi-modal Detection Network. IET Computer Vision, 2024, 18(5), 626-639. <https://doi.org/10.1049/cvi2.12268>
52. Zhong, Y. W., Yang, J. W., Zhang, P. C., Li, C. Y., Codella, N., Li, L. H., Zhou, L. W., Dai, X. Y., Yuan, L., Li, Y., Gao, J. F. RegionCLIP: Region-based Language-Image Pretraining. CVPR 2022, 16772-16782. <https://doi.org/10.1109/CVPR52688.2022.01629>

