Referring Expression Comprehension for Small Objects

Kanoko Goto[1]* Takumi Hirose[1]* Mahiro Ukai[1] Shuhei Kurita[2,1] Nakamasa Inoue[1]

[1]Institute of Science Tokyo [2]National Institute of Informatics

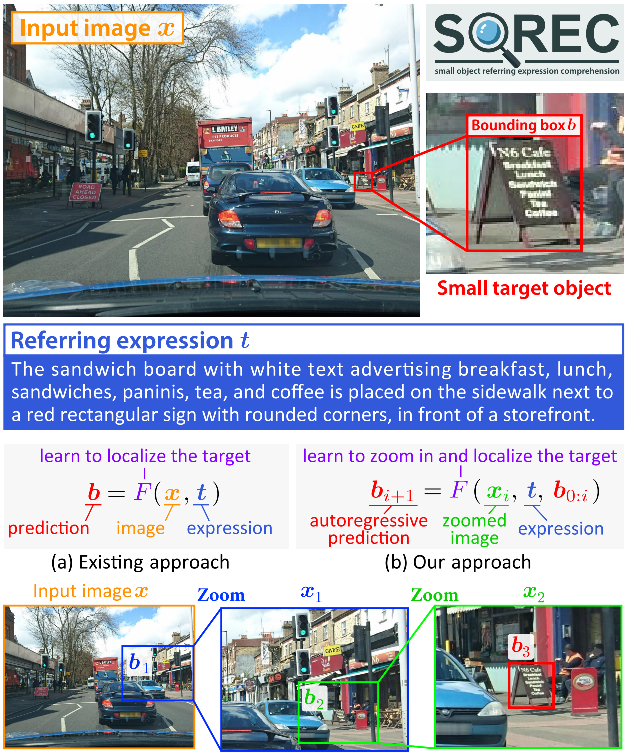
Project Page: https://github.com/mmaiLab/sorec

Abstract

Referring expression comprehension (REC) aims to localize the target object described by a natural language expression. Recent advances in vision-language learning have led to significant performance improvements in REC tasks. However, localizing extremely small objects remains a considerable challenge despite its importance in real-world applications such as autonomous driving. To address this issue, we introduce a novel dataset and method for REC targeting small objects. First, we present the small object REC (SOREC) dataset, which consists of 100,000 pairs of referring expressions and corresponding bounding boxes for small objects in driving scenarios. Second, we propose the progressive-iterative zooming adapter (PIZA), an adapter module for parameter-efficient fine-tuning that enables models to progressively zoom in and localize small objects. In a series of experiments, we apply PIZA to GroundingDINO and demonstrate a significant improvement in accuracy on the SOREC dataset. Our dataset, codes and pre-trained models are publicly available on the project page.

1. Introduction

Object localization in images has been a long-term research topic in the field of computer vision. Early studies introduced image datasets such as Pascal VOC [14] and COCO [39], which involve bounding box annotations for predefined object categories, leading to the development of object detection models including CNN-based models [15, 16, 19, 60] and Transformer-based models [1, 21, 29, 41, 50, 64, 82, 85, 92]. For more detailed and flexible object localization, referring expression comprehension (REC) aims to localize a specific object referred to by a natural language description. REC uses queries like "the red car parked in front of the coffee shop" as input and requires locating this unique object in an input image. RefCOCO, RefCOCO+ and RefCOCOg [53, 78] are the most popular datasets for REC, providing referring expressions for images in the COCO dataset along with Flickr30K entities [57]. Over the past decade, deep learning architectures that bridge visual contents and natural language descriptions have been investigated. Examples include one-stage architectures [38, 48, 62, 75] and two-stage architectures based on CNN-LSTM [24, 40, 49, 52, 54, 78, 79, 83] and attention mechanism [8, 12, 25, 43, 48, 66, 80, 94].


Input image x
SOREC
small object referring expression comprehension
Bounding box b
Small target object
Referring expression t
The sandwich board with white text advertising breakfast, lunch, sandwiches, paninis, tea, and coffee is placed on the sidewalk next to a red rectangular sign with rounded corners, in front of a storefront.
learn to localize the target
$b = F(x, t)$
prediction
image
expression
learn to zoom in and localize the target
$b_{i+1} = F(x_i, t, b_{0:i})$
autoregressive prediction
zoomed image
expression
Input image x
Zoom
x_1
Zoom
x_2
b_1
b_2
b_3

(a) Existing approach (b) Our approach

(c) Autoregressive prediction

Figure 1. The SOREC dataset consists of pairs of referring expressions and bounding boxes for extremely small objects. (a) Existing approach fine-tunes a model F to localize the target. (b) Our approach fine-tunes F to progressively zoom in and localize the target in an autoregressive manner. (c) Example of prediction in three zooming steps.

*Equal contribution

With advances in large-scale vision-language learning, recent models have become capable of precisely understanding object attributes and relations described in natural language [35, 42, 69, 70, 73, 84, 87]. As a result, these models have achieved high accuracy in REC tasks, with accuracy rates of over 90% on the RefCOCO test sets. However, localizing small objects remains a significant challenge. The lack of REC datasets targeting small objects has impeded further progress in this area, despite its critical role in real-world applications such as autonomous driving, where the ability to localize small objects is essential for ensuring safety and facilitating precise decision-making in complex environments.

To address this issue, the present study makes two significant contributions. First, we introduce the small object REC (SOREC) dataset, a new dataset for REC targeting small objects in autonomous driving scenarios. Second, we propose the progressive-iterative zooming adapter (PIZA), an adapter module for parameter-efficient fine-tuning that enables models to progressively zoom in and localize small objects. Below we highlight each contribution.

1) Dataset contribution. We propose the SOREC dataset, which consists of 100,000 pairs of referring expressions and corresponding bounding boxes for extremely small objects in road, highway, rural-area, and off-road scenes. As shown in Figure 1, the typical size of a bounding box is approximately 0.1% of the input image size, making it challenging to localize these objects. To the best of our knowledge, this is the first dataset for REC targeting small objects in autonomous driving scenarios. The availability of this dataset promotes further research advancements in the area.

2) Technical contribution. We propose PIZA, a lightweight learnable module for parameter-efficient fine-tuning. Through fine-tuning with PIZA, the model learns to localize small objects in an autoregressive manner, where a zoomed image is fed into the model iteratively, as shown in Figure 1 (b-c). This approach significantly improves accuracy, as demonstrated in Table 3.

2. Related work

2.1. Tasks and datasets

Referring expression comprehension. Considerable efforts have been dedicated to constructing datasets of referring expressions on images over the past decade. ReferItGame [30] was a pioneering large-scale dataset for REC, consisting of 130k expressions for 20k images collected from ImageCLEF and SAIAPR. RefCOCO, RefCOCO+ [78] and RefCOCOg [53] provided expressions for images in COCO [39], and are the most popular benchmarking datasets. CLEVR-Ref [45] is a diagnostic dataset focusing on compositional language understanding using synthetic images. Refer360 [71] is a dataset for referring expression recognition in 360-degree images. REVERIE [58] offers a dataset for remote embodied visual referring expressions in real indoor environments. RefEgo [33] focuses on egocentric REC in first-person videos.

Small object detection. The importance of small object detection has been recognized in various object detection scenarios. Examples of datasets involving small objects include WiderFace [74] for face detection, TinyPerson [81] for person detection, TT100K [93] for traffic sign detection, VisDrone [91] for drone-based detection, and DOTA [11] for remote sensing detection. SODA [5] is the latest large-scale dataset for small object detection in automatic driving scenarios. Compared with normal-sized object detection datasets such as Pascal VOC [14] and COCO [39], creating datasets for small object detection is generally more expensive as annotating small objects is more challenging.

Other related tasks. Visual grounding (VG) also aims to localize objects given natural language descriptions. In VG, each image may contain multiple target objects, typically described with shorter phrases than those used in REC. Example VG datasets include SK-VG [4] and GigaGrounding [51]. Open-vocabulary object detection aims to detect objects that are not seen in the training dataset. For training open-vocabulary detection models, large object detection datasets are recently used such as O365 [63], GoldG [29], GRIT [56] and V3Det [68].

2.2. Models

REC models have evolved significantly over the years, transitioning from traditional CNN-LSTM architectures to attention and transformer-based architectures. One-stage REC models [38, 48, 62, 75] integrated object detection and language grounding into a unified architecture, allowing for end-to-end training. Two-stage approaches [24, 40, 49, 52, 54, 78, 79, 83] utilized region proposals generated by object detectors and applied LSTM to encode the referring expressions. Attention mechanisms were later incorporated to improve the alignment between image regions and referring expressions [8, 12, 25, 43, 48, 66, 80, 94].

To cover multiple vision tasks recent studies have demonstrated the effectiveness of large-scale vision-language pre-training [7, 36, 42, 69, 70, 73, 84, 87]. For example, open-set detection models such as GLIPv2 [84] and Grounding DINO [42] can handle both object detection and REC. We chose GroundingDINO as the baseline model because it is pre-trained on a union of datasets, including those for relatively small object detection and REC.

2.3. Parameter efficient fine-tuning

Prompt-based fine-tuning. Inspired by prompt tuning methods for natural language processing tasks [27, 44, 65], prompt-based fine-tuning methods have been proposed for computer vision tasks. Context optimization (CoOp) [89]

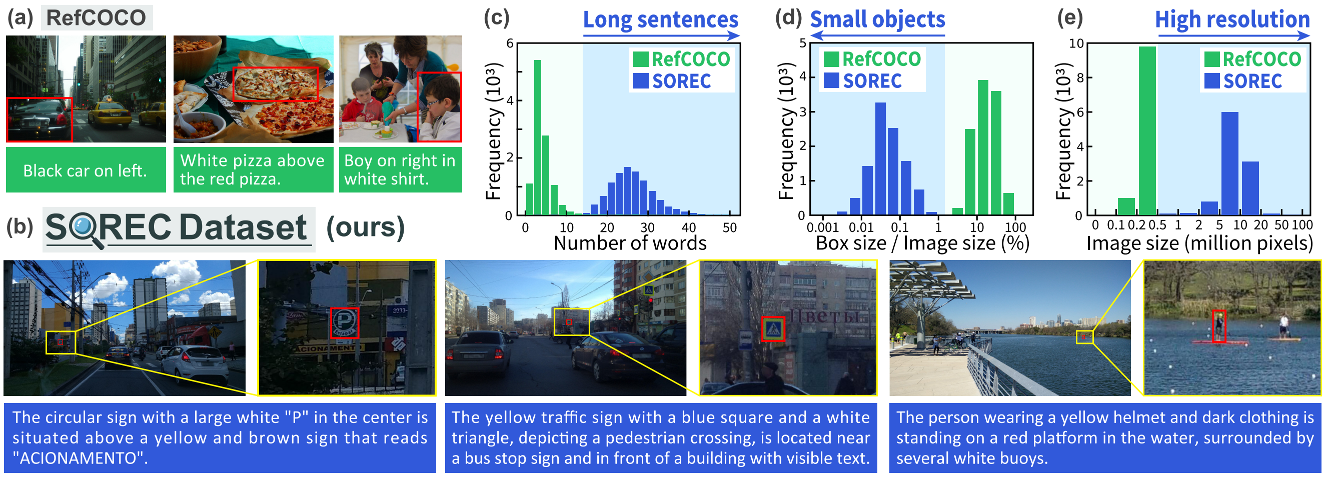


Figure 2. Dataset comparison. (a) RefCOCO is a representative REC dataset consisting of expressions and bounding boxes for normal-sized objects. (d) SOREC is our dataset, consisting of relatively longer expressions compared to RefCOCO, to identify small objects. (c-e) Comparison of word count, image size, and relative bounding box size distributions on test sets.

and visual prompt tuning (VPT) [26] are two representative methods. CoOp incorporated learnable embeddings into the text encoder of CLIP [59]. CoCoOp [88] introduced prompts conditioned by image features. VPT incorporated learnable embeddings into the vision transformers [13]. Further extension includes distribution learning, multi-modal learning and various techniques to leverage pre-trained knowledge [6, 9, 31, 32, 47, 76, 77, 86, 90].

LoRA-based fine-tuning. LoRA [23] introduces low-rank adaptations to the weight matrices of a pre-trained model. LoRA reduces the number of trainable parameters by decomposing the weight updates into low-rank matrices. For further improving parameter efficiency, quantization techniques are also introduced [10, 72].

Adapter-based fine-tuning. Adapters are lightweight learnable modules incorporated into a frozen pretrained model. The first adapter architecture [22] was proposed for Transformers, which inserts adapters into the attention module and the feed forward network module in each encoder layer. There have been substantial efforts in architectural adapter design for various neural networks for computer vision tasks [2, 3, 17, 18, 28, 67]. Adapter+ [67] is a well-designed adapter architecture for vision transformers, which we employ in our experiments.

3. SOREC Dataset

The SOREC dataset consists of 100,000 pairs of referring expressions and corresponding bounding boxes for small objects in road, highway, rural, and off-road images. As shown in Figure 2 (b), referring expressions describe both the characteristics of the target object and its spatial relationships with surrounding objects, in order to locate target objects. Each bounding box typically occupies approximately 0.05% of the entire image area. Compared to existing datasets such as RefCOCO [78] in Figure 2 (a), SOREC presents a particularly challenging task due to the extremely small bounding boxes. This challenge is critical for advancing real-world applications including autonomous driving and surveillance, where detecting small objects is essential.

3.1. Dataset Construction

The SOREC dataset is semi-automatically created in the following five steps.

1) Source selection. We selected the SODA-D dataset [5] as our source dataset. It consists of 24,828 high-quality images for small object detection, collected primarily from the Mapillary Vistas dataset [55]. The average resolution of these images is 3407 × 2470 pixels.

2) Segmentation. To extract small object regions, we applied Semantic-SAM [34] to image patches of size 800 × 800 pixels in a sliding window manner, with the granularity prompt level set to 3. Bounding boxes of object region were also computed. We excluded object regions whose bounding boxes occupied more than 2% of the image.

3) Filtering. Through manual inspection of the extracted object regions, we found many instances of trees and windows that are not suitable for REC. We filtered them out by computing CLIP scores of each bound box region using a prompt of "tree, forest, window." Subsequently, we sorted the object regions based on the score $S \exp(-|a-1|)p$, where S is the bounding box size, a is the aspect ratio and p is the stability score obtained from Semantic-SAM. From the top 200,000 results, we excluded any remaining meaningless objects and selected the top 150,000 results through crowdsourcing.

4) Referring expression generation. For each object region, we cropped an image centered on its bounding box, with a random height between 2.5 and 3.5 times the height

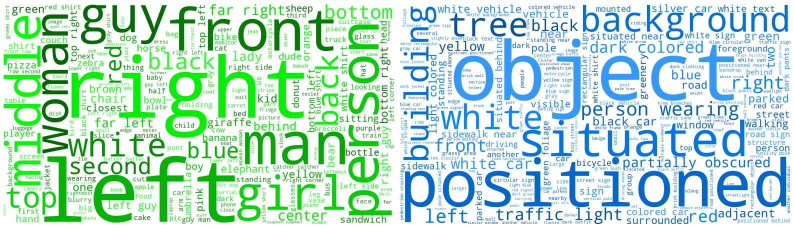


Figure 3. Word clouds for RefCOCO (left) and SOREC (right).

Split	Images	Expressions
Train-S	1,446	10,000
Train-L	13,494	61,369
Validation	2,382	10,712
Test-A	4,107	10,815
Test-B	5,153	17,104

Table 1. Dataset split

of the bounding box, and a width between 1.5 and 3 times its height. We drew a red bounding box on the image with a line thickness of 2 pixels. We input these images into GPT-4o to generate initial referring expressions using the prompt "Write a sentence that refers to the object in the red frame by mentioning its details, colors, relative relationship to surrounding objects." We excluded cases where the generated descriptions did not include references to surrounding objects by counting nouns.

5) Quality control. Finally, using crowdsourcing again, we excluded object regions that could not be uniquely identified by the corresponding expression. This step was a binary decision on whether the object could be identified by the expression, allowing for minor errors in descriptions of surrounding objects, and chose 100,000 high-quality pairs of bounding boxes and expressions. For test sets, we further asked annotators to revise expressions if they involve errors. As a result, 18.45% of sentences were found to contain minor errors related to color, spatial relations, and similar attributes.

3.2. Dataset statistics

Description length. Figure 2 (c) shows the distribution of the number of words per expression. As shown, the average number of words is 25.5, which is approximately seven times longer than the 3.52 words in RefCOCO. This is due to the need for more detailed information to identify small objects, which is a key characteristic of this dataset. Figure 3 compares word distributions by word clouds. As shown, SOREC dataset contains words related to positioning, such as 'positioned' and 'situated'.

Bounding box size. Figure 2 (d) shows the distribution of bounding box sizes relative to the image sizes. As indicated, all the target bounding boxes occupy less than 1% of the image area, presenting a challenging REC task. Since pre-training is often performed on datasets that feature normal sized objects, fine-tuning is necessary to bridge this gap for localizing small target object.

Image size. Figure 2 (e) shows the distribution of image sizes. As shown, the SOREC dataset consists of high-resolution images that are sufficient for performing REC targeting small objects.

Data split. We created training, validation, test splits as summarized in Table 1. The train-L set is the full training set, and the train-S set is a small subset consisting of 10,000 expressions. The validation set consists of 10,712 expressions, with no overlap in images with the training sets. The test-A and -B sets contain expressions for traffic objects and the other objects.

4. Method

This section describes PIZA, a lightweight adapter module for parameter-efficient fine-tuning that enables models to localize target small objects by progressively and iteratively zooming into them.

4.1. Preliminary

Problem settings. Let $\boldsymbol{x} \in \mathbb{R}^{W\times H\times C}$ be an input image, where W is the width, H is the height, and C is the number of channels. We denote by $\boldsymbol{b} = (x_0, y_0, x_1, y_1) \in \mathbb{R}^4$ a bounding box, where (x_0, y_0) is the coordinate of the top-left corner, and (x_1, y_1) is the coordinate of the bottom-right corner of the box. Given a natural language expression $\boldsymbol{t}$, the goal of REC is to localize the object corresponding to $\boldsymbol{t}$ in the image. As such, the model F takes as input $(\boldsymbol{x}, \boldsymbol{t})$ and learns to predict a bounding box as $\hat{\boldsymbol{b}} = F(\boldsymbol{x}, \boldsymbol{t})$, so that the predicted bounding box $\hat{\boldsymbol{b}}$ matches the ground truth bounding box $\boldsymbol{b}^*$. This work focuses on the setting where the size of $\boldsymbol{b}^*$ is significantly smaller than the image size, *i.e.*, we assume that $|\boldsymbol{b}^*| \ll WH$, where $|\boldsymbol{b}| = (x_1 - x_0)(y_1 - y_0)$ indicates the size of $\boldsymbol{b}$.

Vision-language pre-training. We assume that a pre-trained model F is given and explore parameter-efficient fine-tuning methods by which the model quickly adapts to localize small objects.

Difficulty. The main difficulty lies in the gap between pre-training and fine-tuning regarding the bounding box sizes and expression lengths. Recent object localization models, such as GroundingDINO [42], are capable of both REC and open-set object detection because they are pre-trained on a large union set of REC and object detection datasets. However, a challenge arises when dealing with a combination of long sentences and extremely small objects. We aim to address this challenge in parameter-efficient fine-tuning.

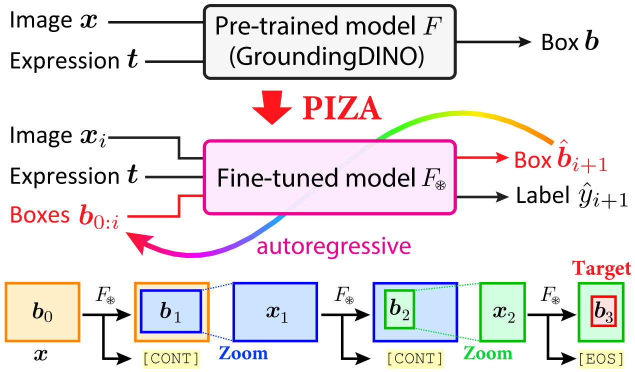


Figure 4. **Fine-tuning with PIZA.** Given a pre-trained model F, PIZA produces a model $F_{\circledast}$ that zooms in to localize small objects in an autoregressive manner through fine-tuning. In the inference phase, bounding boxes $\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_T$ indicating zooming steps are predicted to localize the target at the end.

4.2. Progressive-integrative zooming adapter

Overview. Zooming in to localize a small target object can be understood as a search problem over an image. We model a search process P as a sequence of bounding boxes:

$$P = (\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_T), \qquad (1)$$

where $\boldsymbol{b}_0 = (0, 0, W, H)$ indicates the bounding box covering the entire image, $\boldsymbol{b}_T$ is the final small bounding box for localizing the target, T is the number of zooming steps, and $\boldsymbol{b}_i$ covers $\boldsymbol{b}_j$ if $i < j$, as shown in Figure 1 (c).

Through fine-tuning, models learn to predict search processes so that the final bounding box matches the ground truth, *i.e.*, $\boldsymbol{b}_T \simeq \boldsymbol{b}^*$. To this end, PIZA extends the pre-trained model F to a model $F_{\circledast}$ that predicts a search process in an autoregressive manner as

$$\hat{\boldsymbol{b}}_{i+1} = F_{\circledast}(\boldsymbol{x}_i, \boldsymbol{t}, \boldsymbol{b}_{0:i}), \qquad (2)$$

where $\boldsymbol{x}_i$ is the cropped image region corresponding to $\boldsymbol{b}_i$, $\boldsymbol{t}$ is an input expression, $\boldsymbol{b}_{0:i} = (\boldsymbol{b}_0, \cdots, \boldsymbol{b}_i)$ is a subsequence of bounding boxes and $\circledast$ indicates that PIZA is applied. A visualization of this procedure is shown in Figure 4. Since F is a function that accepts two inputs, $\boldsymbol{x}$ and $\boldsymbol{t}$, $F_{\circledast}$ is built up by incorporating a module that can take $\boldsymbol{b}_{0:i}$ as an additional input into F. Below we describe the module architecture.

PIZA module. Inspired by time-step embeddings in diffusion models [20, 61] that represent stages of the diffusion process, the PIZA module learns zooming-step embeddings that represent progress of the search process. Specifically, the zooming-step embedding $\boldsymbol{h} \in \mathbb{R}^d$ is extracted from the input sequence of bounding boxes $\boldsymbol{b}_{0:i} \in \mathbb{R}^{4\times(i+1)}$ in two steps. First, a sequence of low-level features $\boldsymbol{l}_{0:i} = (\boldsymbol{l}_0, \boldsymbol{l}_1, \cdots, \boldsymbol{l}_i)$ is extracted. Each feature $\boldsymbol{l}_j$ is a 6 dimensional vector, and its elements are listed in Table 2. Second, $\boldsymbol{h}$ is extracted by feeding $\boldsymbol{l}_{0:i}$ into a small learnable module. Figure 5 shows the architecture consisting of a sequence of learnable Fourier embeddings [37], a transformer encoder and an average pooling layer. The embeddings are trained with two heads: an EOS head and a progress head. The EOS head predicts a binary label $\hat{y}_{i+1} \in \{\texttt{[CONT]}, \texttt{[EOS]}\}$ indicating either "continue to search (CONT)" or "end of search (EOS)". The progress head predicts progress of search $\hat{z}_{i+1} \in [0, 1]$ expressed as a real value, where 0.0 indicates the start and 1.0 indicates the end of the search process. The binary cross-entropy loss and mean squared error loss are applied to these heads, respectively, on the extended training dataset described in Section 4.4. In the inference phase, search is stopped when the EOS label is predicted. The number of parameters of this module is 0.27M and the feature dimension is set to 16. The detailed architectural hyperparameters are provided with our code.

Feature	Definition
Normalized size	$s_j = \lvert\boldsymbol{b}_j\rvert / \lvert\boldsymbol{b}_0\rvert$
Relative size	$r_0 = 1,\ r_j = \lvert\boldsymbol{b}_j\rvert / \lvert\boldsymbol{b}_{j-1}\rvert$
Normalized width	$w_j = (x_1^{(j)} - x_0^{(j)})/W$
Normalized height	$h_j = (h_1^{(j)} - h_0^{(j)})/H$
Center position (x-axis)	$\bar{x}_j = (x_0^{(j)} + x_1^{(j)})/(2W)$
Center position (y-axis)	$\bar{y}_j = (y_0^{(j)} + y_1^{(j)})/(2H)$

Table 2. Low-level features extracted from the sequence of bounding boxes $\boldsymbol{b}_j = (x_0^{(j)}, x_1^{(j)}, y_0^{(j)}, y_1^{(j)})$. W and H denotes the width and height of the input image.

4.3. Parameter-efficient fine-tuning with PIZA

To perform fine-tuning with the PIZA module, we incorporate the embeddings $\boldsymbol{h}$ into parameter-efficient fine-tuning methods. Here, we propose prompt-based, LoRA-based and adapter-based fine-tuning with PIZA. Their architectures are shown in Figure 6.

PIZA-CoOp. CoOp [89] is a prompt-based fine-tuning method, which prepends learnable embeddings to the input text prompts as $G(\boldsymbol{x}, \boldsymbol{t}) = F(\boldsymbol{x}, [\boldsymbol{e}, \boldsymbol{t}])$, where $\boldsymbol{e} = (\boldsymbol{e}_0, \boldsymbol{e}_1, \cdots, \boldsymbol{e}_L)$ is a sequence of learnable embeddings. PIZA-CoOp inserts $\boldsymbol{h}$ as $G_{\circledast}(\boldsymbol{x}, \boldsymbol{t}) = F(\boldsymbol{x}, [\boldsymbol{e}, H(\boldsymbol{h}), \boldsymbol{t}])$, where H is a learnable linear layer as shown in Figure 6 (a).

PIZA-LoRA. LoRA [23] is a low-rank adaptation method that injects trainable low-rank matrices into linear projections as $W\boldsymbol{x} + BA\boldsymbol{x}$, where $\boldsymbol{x}$ is an input, W is a frozen weight matrix and A, B are learnable low-rank matrices. PIZA-LoRA integrates the embeddings $\boldsymbol{h}$ into the bottleneck of LoRA as shown in Figure 6 (b), resulting in $W\boldsymbol{x} + BA\boldsymbol{x} + BC\boldsymbol{h}$, where C is a newly added learnable matrix. We set the rank to 16.

PIZA-Adapter+. Adapter+ [67] utilizes the post-adapter architecture, channel-wise scaling and Houlsby initialization [22]. PIZA-Adapter+ adds the embeddings $\boldsymbol{h}$ to the output of the channel-wise scaling layer as shown in Fig-

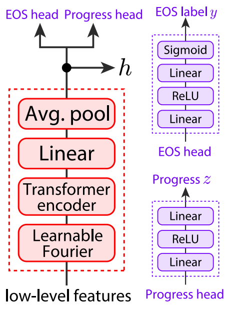
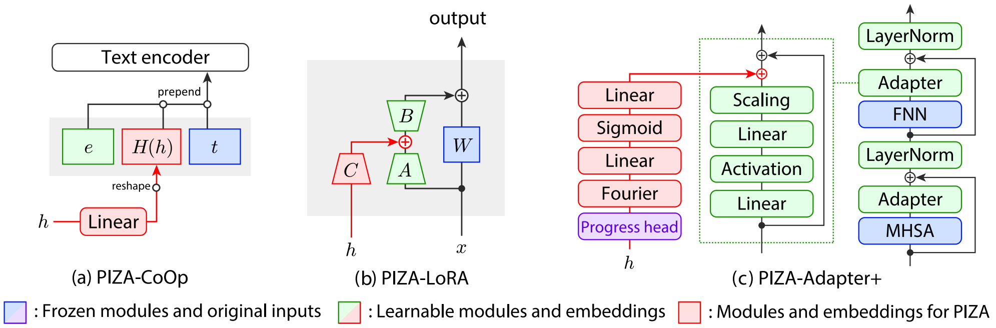


Figure 5. PIZA module.



(a) PIZA-CoOp (b) PIZA-LoRA (c) PIZA-Adapter+

Figure 6. Parameter efficient fine-tuning with PIZA.

ure 6 (c). We set the bottleneck dimension to 256.

4.4. Training

Given a training dataset $\mathcal{D}$ consisting of images, expressions and ground truth bounding boxes, we construct an extended dataset $\mathcal{E}$ that involves ground truth search processes for training with PIZA.

Overview. Each ground truth search process P^* represents zooming steps to localize the target object. Specifically, it is given by $P^* = (\boldsymbol{b}_0^*, \boldsymbol{b}_1^*, \cdots, \boldsymbol{b}_{T^*}^*)$, where $\boldsymbol{b}_0^* = (0, 0, W, H)$ indicates the entire image and $\boldsymbol{b}_{T^*}^* = \boldsymbol{b}^*$ is the ground truth bounding box. To facilitate efficient fine-tuning, we generate P^* so that the distribution of inverse zoom factors $(z_j^*)^{-1} = |\boldsymbol{b}_j^*|/|\boldsymbol{b}_{j-1}^*|$ match to the distribution of bounding box area ratios $p(r)$ in pre-training.

Distribution $\boldsymbol{p(r)}$. Along with a pre-trained model, the distribution $p(r)$ is pre-estimated by applying kernel density estimation to the pre-training dataset (we use the union of O365 and GoldG in our experiments). We randomly sample 100,000 bounding boxes to compute area ratios $r = |\boldsymbol{b}^*|/(WH)$ and apply a Gaussian kernel.

Width and height for $\boldsymbol{b}_j^*$. The width and height for each bounding box in P^* are randomly sampled in three steps. First, ratios r_k are randomly sampled from $p(r)$ for $k = 1, 2, \cdots, T_{\max}$, where $T_{\max}$ is a sufficiently large constant. Second, the number of zooming steps T^* is determined such that the cumulative product of r_k is closest the area ratio r^* of the ground truth bounding box:

$$T^* = \underset{T}{\operatorname{argmin}} \left(\frac{1}{r^*} \prod_{k=1}^{T} r_k - 1 \right), \; r^* = \frac{|\boldsymbol{b}^*|}{WH}, \tag{3}$$

where $\boldsymbol{b}^*$ is a bounding box sampled from $\mathcal{D}$, and W, H are the width and height of the corresponding image. Then, the zoom factors z_j^* and the size of bounding boxes S_j^* are determined as follow:

$$z_j^* = \left(\frac{1}{r^*} \prod_{k=1}^{T^*} r_k^{\omega_k^{-1}} \right)^{\frac{1}{T^*}} r_j^{-1}, \; S_j^* = \left(\prod_{k=1}^{T^*-j} z_{T^*-k} \right) |\boldsymbol{b}^*|, \tag{4}$$

where $\omega_k = \lambda_1 e^{-\lambda_2 k} / \sum_{k'=1}^{T^*} \lambda_1 e^{-\lambda_2 k'}$ are weights derived from an exponential distribution. This weighting ensures that $z_j^* \simeq r_j^{-1}$ with smaller j, encouraging more precise bounding box predictions at the initial zooming step. Finally, the width and height of $\boldsymbol{b}_j^*$ are determined by $w_j^* = a_j \sqrt{S_j^*}, \quad h_j^* = a_j^{-1} \sqrt{S_j^*}$ where a_j is an interpreted aspect ratio

$$a_j = \frac{W}{H} - \left(\frac{W}{H} - 1 \right) \frac{WH - S_j^*}{WH - S_{T^*-1}^*}. \tag{5}$$

Centers. The center of each bounding box is aligned with the center coordinates of the target object. If the region extends beyond the image boundaries, the bounding box is minimally shifted to remain entirely within the image.

Labels. Finally, for the generated search process P^*, the sequence of binary labels $\boldsymbol{y}^* = (y_0^*, y_1^*, \cdots, y_{T^*}^*)$ and zooming step labels $z = (z_0^*, z_1^*, \cdots, z_{T^*}^*,)$ are attached. Each label is given by

$$y_j^* = \begin{cases} \texttt{[CONT]} & (0 \le j < T^*) \\ \texttt{[EOS]} & (j = T^*) \end{cases}, \quad z_j^* = \frac{j}{T^*} \tag{6}$$

Loss function. The loss for fine-tuning is computed in three steps. First, a mini-batch of quadruplets $(\boldsymbol{x}, \boldsymbol{t}, P^*, \boldsymbol{y}^*)$ is drawn from $\mathcal{E}$. Second, for each quadruplet, index $i \in \{1, 2, \cdots, T^* - 1\}$ is randomly drawn to compute the forward process $\hat{\boldsymbol{b}}_{i+1} = F_{\circledast}(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{b}_{0:i})$. Finally, the loss depending on the pre-trained model is applied. In the experiments, we implement PIZA over the GroundingDINO model; thus the loss consists of the contrastive loss and the localization loss [42].

5. Experiments

5.1. Experimental settings

Dataset and metrics. The SOREC dataset is used for training and evaluation. We report the mean accuracy (mAcc) over IoU thresholds from 0.50 to 0.95 in increments of 0.05, as well as the accuracy at IoU of 0.50 (Acc_{50}) and 0.75 (Acc_{75}).

Method	#Params	Train-S									Train-L								
		Val			Test-A			Test-B			Val			Test-A			Test-B		
		mAcc	Acc_{50}	Acc_{75}	mAcc	Acc_{50}	Acc_{75}	mAcc	Acc_{50}	Acc_{75}	mAcc	Acc_{50}	Acc_{75}	mAcc	Acc_{50}	Acc_{75}	mAcc	Acc_{50}	Acc_{75}
Zero-shot	0	0.2	0.6	0.0	0.3	1.0	0.1	0.0	0.2	0.0	0.2	0.6	0.0	0.3	1.0	0.1	0.0	0.2	0.0
Full fine-tuning	173.0M	29.5	51.7	30.2	35.9	58.6	38.8	23.0	43.6	21.9	37.4	$\underline{63.7}$	39.2	43.8	$\underline{69.6}$	48.0	30.5	$\underline{55.6}$	29.8
CoOp	0.1M	20.2	36.1	20.4	24.2	40.1	25.8	15.5	29.6	14.6	22.6	41.6	22.0	27.5	46.5	28.7	17.5	34.8	15.9
PIZA-CoOp (Ours)	0.9M	26.3	39.1	29.7	29.4	41.2	34.2	21.9	33.8	24.3	29.8	44.1	33.5	33.4	46.8	38.7	24.4	37.6	26.9
LoRA	1.3M	21.6	38.5	21.8	26.2	43.1	28.1	17.0	32.5	15.9	25.2	44.5	25.3	30.7	50.2	33.0	19.7	37.3	18.8
PIZA-LoRA (Ours)	1.5M	30.9	44.7	34.9	33.8	46.6	39.2	25.8	38.7	28.9	34.5	49.9	39.1	39.3	54.0	45.5	29.0	43.4	32.4
Adapter+	3.3M	26.0	48.1	24.8	32.0	55.0	33.3	20.3	40.4	17.9	34.6	59.5	35.7	40.7	65.9	44.4	27.6	51.3	26.6
PIZA-Adapter+ (Ours)	3.5M	$\underline{36.8}$	$\underline{53.5}$	$\underline{41.8}$	$\underline{43.1}$	$\underline{59.6}$	$\underline{50.1}$	$\underline{30.4}$	$\underline{45.9}$	$\underline{34.1}$	$\underline{39.0}$	60.6	$\underline{42.9}$	$\underline{45.1}$	66.2	$\underline{51.7}$	$\underline{31.7}$	52.2	$\underline{33.6}$

Table 3. Parameter-efficient fine-tuning results. #Params indicates the number of fine-tuned parameters. Best results are underlined.

Pre-trained model. We selected GroundingDINO [42] using Swin-T [46] as a baseline model, and used the improved version provided as MM-GroundingDINO in the mmdetection library [87]. This model is pre-trained on the union of the following four datasets: O365 [63], GoldG [29], GRIT [56] and V3Det [68].

Baselines. We implemented three baselines for parameter-efficient fine-tuning: CoOp [89], LoRA [23] and Adapter+ [67]. PIZA is applied to each method as described in Section 4.3. Zero-shot baseline results are also reported.

Implementation details. The AdamW optimizer is used for 5 epochs with a learning rate of 2×10^{-4}, which is decayed by a factor of 0.5 at epoch 3. The hyperparameters for AdamW are set to their default values in PyTorch. The batch size is set to 16. LoRA is applied to each self-attention and cross-attention module. Adapter+ modules are inserted after each self-attention and feed-forward network module. Further details are provided in Appendix.

5.2. Experimental results

Main results. Table 3 summarizes the parameter-efficient fine-tuning results. As shown, PIZA significantly improved the performance for all methods. PIZA-Adapter+ achieved the best performance in terms of mAcc, surpassing the full fine-tuning baseline while reducing the number of learnable parameters from 173.0M to 3.5M. Prompt-tuning methods (CoOp and PIZA-CoOP) were more efficient but less effective than Adapter+ and LoRA methods. This is likely due to the low performance of the zero-shot baseline, which suggests that prompt-tuning alone may struggle to bridge the gap between the vision-language pre-training task and the REC task for small objects. When comparing Test-A and Test-B, all models exhibited higher accuracy on Test-A, which consists of traffic objects. This result is understandable, as objects such as traffic lights and road signs are designed with colors and shapes that make them easily to detect. When comparing training dataset sizes, the larger dataset (Train-L) consistently demonstrated higher performance, suggesting that further increasing the dataset size could be beneficial.

Method	#Prm.	Val	Test-A	Test-B
PIZA-Adapter+	3.5M	$\underline{36.8}$/$\underline{53.5}$/$\underline{41.8}$	$\underline{43.1}$/$\underline{59.6}$/$\underline{50.1}$	$\underline{30.4}$/$\underline{45.9}$/$\underline{34.1}$
w/o emb. insertion	3.5M	36.7/53.2/41.7	42.8/59.2/49.9	30.3/45.8/34.0
w/o PIZA module	3.3M	26.0/48.1/24.8	32.0/55.0/33.3	20.3/40.4/17.9
$d = 256$	3.5M	$\underline{36.8}$/$\underline{53.5}$/$\underline{41.8}$	$\underline{43.1}$/$\underline{59.6}$/$\underline{50.1}$	$\underline{30.4}$/$\underline{45.9}$/34.1
$d = 128$	2.4M	36.4/52.8/41.6	41.8/57.9/48.5	30.3/45.5/$\underline{34.2}$
$d = 64$	1.9M	36.6/52.9/$\underline{41.8}$	42.2/58.1/49.0	29.9/45.1/33.6
$d = 32$	1.6M	35.1/51.0/40.1	40.8/56.3/47.4	29.0/43.7/32.5

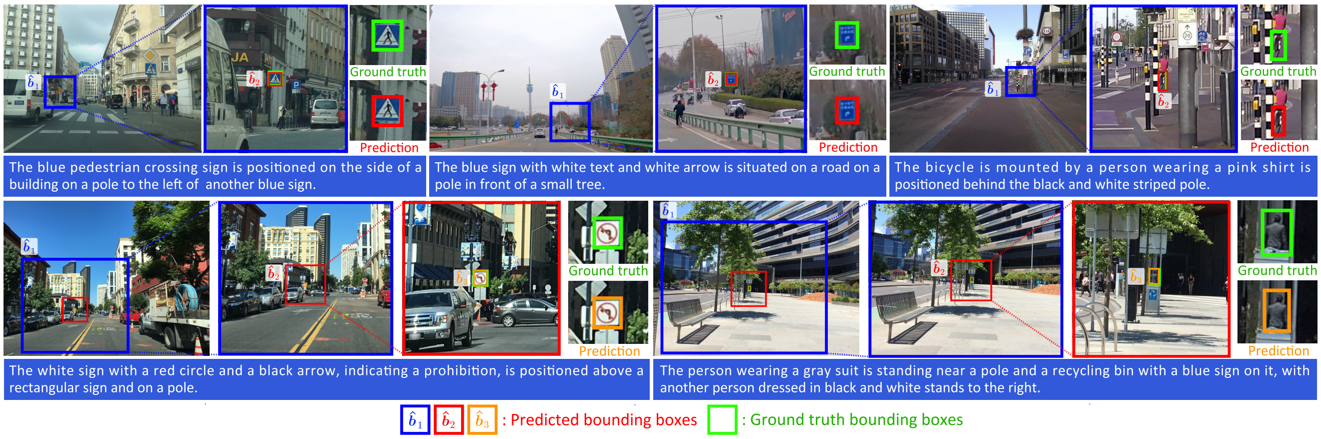
Table 4. Ablation and hyperparameter studies for PIZA-Adapter+. "w/o embedding insertion" omits the connection colored in red in Figure 6 (c). d is the bottleneck dimension of the adapter. Each triplet of values indicates mAcc/Acc_{50}/Acc_{75}. Train-S is used for training.

Ablation and hyperparameter studies. Tables 4 and 5 show the results of ablation and hyperparameter studies for PIZA-Adapter+, PIZA-LoRA, and PIZA-CoOp. As shown, autoregressive prediction with the PIZA module is essential, and incorporating the zooming-step embedding further improved the performance. For PIZA-Adapter+, performance improves as the bottleneck dimensions increase.

Necessity of pre-training. Table 6 compares training from scratch and full fine-tuning. Although the zero-shot baseline performance was low, the results confirmed that pre-training is necessary.

Zooming steps. Table 7 presents the tradeoff between the number of zooming steps and performance by comparing our best results, which resulted in 2.11 steps on average, with those obtained by enforcing the number of steps T^* to 1, 2 and 3 when creating the extended training dataset. As shown, our method performed the best among the tested configurations. Some qualitative examples are shown in Figure 7. As shown, for the SOREC dataset, 2 or 3 zooming steps were sufficient in most cases.

Comparison with greedy approaches. To validate the necessity of the multi-step inference, we compare PIZA with sliding window and grid-like separation approaches. Figure 8 (a) compares our method with the eight fully


$\hat{b}_1$
$\hat{b}_2$
$\hat{b}_3$
JA
Ground truth
Prediction
$\hat{b}_1$ $\hat{b}_2$ $\hat{b}_3$: Predicted bounding boxes
: Ground truth bounding boxes

The blue pedestrian crossing sign is positioned on the side of a building on a pole to the left of another blue sign.

The blue sign with white text and white arrow is situated on a road on a pole in front of a small tree.

The bicycle is mounted by a person wearing a pink shirt is positioned behind the black and white striped pole.

The white sign with a red circle and a black arrow, indicating a prohibition, is positioned above a rectangular sign and on a pole.

The person wearing a gray suit is standing near a pole and a recycling bin with a blue sign on it, with another person dressed in black and white stands to the right.

Figure 7. Qualitative examples.

Method	#Prm.	Val	Test-A	Test-B
PIZA-LoRA	1.5M	30.9/44.7/34.9	33.8/46.6/39.2	25.8/38.7/28.9
w/o emb. insertion	1.5M	30.2/43.9/34.0	33.5/46.4/38.6	25.3/38.1/28.3
w/o PIZA module	1.3M	21.6/38.5/21.8	26.2/43.1/28.1	17.0/32.5/15.9
PIZA-CoOp	0.9M	26.3/39.1/29.7	29.4/41.2/34.2	21.9/33.8/24.3
w/o emb. insertion	0.3M	26.1/38.7/29.0	29.3/40.9/33.7	21.6/33.2/23.9
w/o PIZA module	0.1M	20.2/36.1/20.4	24.2/40.1/25.8	15.5/29.6/14.6

Table 5. Ablation study for PIZA-LoRA and PIZA-CoOp (Train-S). Each triplet of values indicates mAcc/Acc$_{50}$/Acc$_{75}$.

Method	#Prm.	Val	Test-A	Test-B
Scratch	173M	0.00/0.00/0.00	0.00/0.01/0.00	0.00/0.00/0.00
Full fine-tuning	173M	29.5/51.7/30.2	35.9/58.6/38.8	23.0/43.6/21.9

Table 6. Comparison with training from scratch (Train-S).

Method	Steps	Val	Test-A	Test-B
PIZA-Adapter+	2.11	36.8/53.5/41.8	43.1/59.6/50.1	30.4/45.9/34.1
w/ step enforcing	1.0	25.8/47.5/25.5	31.7/54.2/33.3	20.0/39.3/18.1
w/ step enforcing	2.0	36.1/53.1/40.8	41.7/57.8/48.6	29.6/45.6/32.9
w/ step enforcing	3.0	34.3/50.3/39.0	39.8/55.3/46.3	27.1/41.8/30.1

Table 7. Zooming step analysis (Train-S).

fine-tuned sliding window baselines using combinations of four window sizes $W \in \{500, 750, 1000, 2000\}$ and two strides $S \in \{W, W/2\}$. Apparently, the sliding window approach can improve the performance. However, all of these baselines indeed yield significantly lower performance and higher computational cost than our method. Although smaller windows and smaller strides capture finer details, they exponentially increase computational cost and often lead to false positive detections. Our method addresses these limitations, being 7.3× faster and using 49.4× fewer learnable parameters than the best sliding window baseline. Figure 8 (b) shows comparison with the tile-grid baselines. Our method outperformed them for the same reason discussed for the sliding window approach, suggesting that our method with average zooming step of 2.11 is reasonable compared to these greedy grid-like separation counterparts. The typical window size used in the first step of our method was around 500×500, while that for the tile grid approach on a typical high-resolution image in our dataset was 3407×2470, resulting in a 36-fold increase in inference cost for the greedy tile grid approach.

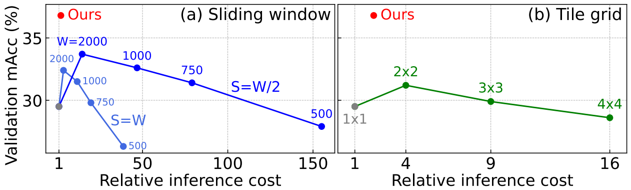


Fig. 8.Comparison with sliding window and tile-grid baselines.

6. Conclusion

We introduced the SOREC dataset, a new dataset for referring expression comprehension targeting small objects. Furthermore, we proposed PIZA tuning, a novel parameter-efficient fine-tuning approach that allows models to progressively zoom in and localize small objects based on natural language expressions.

Future work and limitations. This work focused on localizing small objects in autonomous driving scenarios since detecting such objects is critical for ensuring safety and improves the overall reliability. Extending the proposed dataset to include more diverse environments and object types would remain an interesting future research direction. In addition, extending this work to video data and applying PIZA tuning to architectures for video processing would also be a promising next step. We believe that this work contributed to the computer vision community from both dataset and technical perspectives.

Acknowledgment. This work was supported by JSPS KAKENHI 25K03135, 23H00490, JST CRONOS JPMJCS24K6, and JST BOOST Program JPMJBY24C5, and was carried out using the TSUBAME4.0 supercomputer at Institute of Science Tokyo.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. European Conference on Computer Vision (ECCV)*, pages 213–229, 2020. 1

[2] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 25428–25440, 2022. 3

[3] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *Proc. International Conference on Learning Representations (ICLR)*, 2023. 3

[4] Zhihong Chen, Ruifei Zhang, Yibing Song, Xiang Wan, and Guanbin Li. Advancing visual grounding with scene knowledge: Benchmark and method. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15039–15049, 2023. 2

[5] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023. 2, 3

[6] Eulrang Cho, Jooyeon Kim, and Hyunwoo J Kim. Distribution-aware prompt tuning for vision-language models. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 22004–22013, 2023. 3

[7] Ming Dai, Lingfeng Yang, Yihao Xu, Zhenhua Feng, and Wankou Yang. SimVG: A Simple Framework for Visual Grounding with Decoupled Multi-modal Fusion. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024. 2

[8] Chao Deng, Qi Wu, Qingyao Wu, Fei Hu, Feng Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7746–7755, 2018. 1, 2

[9] Mohammad Mahdi Derakhshani, Enrique Sanchez, Adrian Bulat, Victor Guilherme Turrisi da Costa, Cees GM Snoek, Georgios Tzimiropoulos, and Brais Martinez. Bayesian prompt learning for image-language model generalization. *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 3

[10] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 3

[11] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge J Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(11):7778–7796, 2022. 2

[12] Yang Ding, Jing Yu, Bang Liu, and Yue Hu. Revisiting counterfactual problems in referring expression comprehension. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1, 2

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[14] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision (IJCV)*, 88(2):303–338, 2010. 1, 2

[15] Ross Girshick. Fast r-cnn. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1440–1448. IEEE, 2015. 1

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014. 1

[17] Tianxiang Hao, Hui Chen, Yuchen Guo, and Guiguang Ding. Consolidator: Mergeable adapter with grouped connections for visual adaptation. In *Proc. International Conference on Learning Representations (ICLR)*, 2023. 3

[18] Haoyu He, Jianfei Cai, Jing Zhang, Dacheng Tao, and Bohan Zhuang. Sensitivity-aware visual parameter-efficient fine-tuning. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11791–11801, 2023. 3

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020. 5

[21] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, and Badong Chen. Salience detr: Enhancing detection transformer with hierarchical salience filtering refinement. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17574–17583, 2024. 1

[22] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *Proc. International Conference on Machine Learning (ICML)*, pages 2790–2799, 2019. 3, 5

[23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proc. International Conference on Learning Representations (ICLR)*, 2022. 3, 5, 7

[24] Ronghang Hu, Huijuan Xu, Marcus Rohrbach, Jiashi Feng, Kate Saenko, and Trevor Darrell. Natural language object retrieval. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4555–4564, 2016. 1, 2

[25] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1115–1124, 2017. 1, 2

[26] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *Proc. European Conference on Computer Vision (ECCV)*, 2022. 3

[27] Zhengbao Jiang, Frank F Xu, Junichi Araki, and Graham Neubig. How can we know what language models know? In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5191–5200, 2020. 2

[28] Shibo Jie and Zhi-Hong Deng. FacT: Factor-tuning for lightweight adaptation on vision transformer. In *Proc. AAAI Conference on Artificial Intelligence*, pages 1060–1068, 2023. 3

[29] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR-modulated detection for end-to-end multi-modal understanding. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1780–1790, 2021. 1, 2, 7

[30] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, 2014. 2

[31] Muhammad Uzair Khattak, Hanoona Rasheed, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Maple: Multi-modal prompt learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19113–19122, 2023. 3

[32] Muhammad Uzair Khattak, Syed Talal Wasim, Muzammal Naseer, Salman Khan, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Self-regulating prompts: Foundational model adaptation without forgetting. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15190–15200, 2023. 3

[33] Shuhei Kurita, Naoki Katsura, and Eri Onami. Refego: Referring expression comprehension dataset from first-person perception of ego4d. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3421–3431, 2023. 2

[34] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. Semantic-SAM: Segment and recognize anything at any granularity. In *Proc. European Conference on Computer Vision (ECCV)*, 2024. 3

[35] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, 2022. 2

[36] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[37] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. Learnable fourier features for multi-dimensional spatial positional encoding. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 15816–15829, 2021. 5

[38] Yunhua Liao, Shuang Liu, Guanbin Li, Fei Wang, Yunchao Chen, Chen Qian, and Bo Li. A real-time cross-modality correlation filtering method for referring expression comprehension. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10877–10886, 2020. 1, 2

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 1, 2

[40] Jiajun Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4856–4864, 2017. 1, 2

[41] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. DAB-DETR: Dynamic anchor boxes are better queries for DETR. In *Proc. International Conference on Learning Representations (ICLR)*, 2022. 1

[42] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proc. European Conference on Computer Vision (ECCV)*, 2024. 2, 4, 6, 7

[43] Xiaohui Liu, Zhizhong Wang, Jing Shao, Xiaogang Wang, and Hongsheng Li. Improving referring expression grounding with cross-modal attention-guided erasing. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1950–1959, 2019. 1, 2

[44] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 61–68, 2022. 2

[45] Yining Liu, Somak Aditya, and Yonatan Bisk Lee. Clevr-ref+: Diagnosing visual reasoning with referring expressions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[46] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, 2021. 7

[47] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[48] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collabora-

tive network for joint referring expression comprehension and segmentation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10034–10043, 2020. 1, 2

[49] Renjie Luo and Greg Shakhnarovich. Comprehension-guided referring expressions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7102–7111, 2017. 1, 2

[50] Wenyu Lv, Shangliang Xu, Yian Zhao, Guanzhong Wang, Jinman Wei, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[51] Tao Ma, Bing Bai, Haozhe Lin, Heyuan Wang, Yu Wang, Lin Luo, and Lu Fang. When visual grounding meets gigapixel-level large-scale scenes: Benchmark and approach. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22119–22128, 2024. 2

[52] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2016. 1, 2

[53] Jiayuan Mao, Jing Huang, Alexander Toshev, Oana-Maria Camburu, Alan Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–20, 2016. 1, 2

[54] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *Proc. European Conference on Computer Vision (ECCV)*, pages 792–807, 2016. 1, 2

[55] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5000–5009, 2017. 3

[56] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*, 2023. 2, 7

[57] Bryan A. Plummer, Liwei Wang, Christopher M. Cervantes, Juan C. Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2641–2649, 2015. 1

[58] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in real indoor environments. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9982–9991, 2020. 2

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021. 3

[60] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015. 1

[61] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 5

[62] Arka Sadhu, Kan Chen, and Ram Nevatia. Zero-shot grounding of objects from natural language queries. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4694–4703, 2019. 1, 2

[63] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2, 7

[64] Liu Shilong, Liang Yaoyuan, Huang Shijia, Li Feng, Zhang Hao, Su Hang, Zhu Jun, and Zhang Lei. DQ-DETR: Dual query detection transformer for phrase extraction and grounding. In *Proc. AAAI Conference on Artificial Intelligence*, 2023. 1

[65] Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, 2020. 2

[66] Sijie Song, Xudong Lin, Jiaying Liu, Zongming Guo, and Shih-Fu Chang. Co-grounding networks with semantic attention for referring expression comprehension in videos. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1346–1355, 2021. 1, 2

[67] Jan-Martin O. Steitz and Stefan Roth. Adapters strike back. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23449–23459, 2024. 3, 5, 7

[68] Jiaqi Wang, Pan Zhang, Tao Chu, Yuhang Cao, Yujie Zhou, Tong Wu, Bin Wang, Conghui He, and Dahua Lin. V3det: Vast vocabulary visual detection dataset. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19844–19854, 2023. 2, 7

[69] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. OFA: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proc. International Conference on Machine Learning (ICML)*, pages 23318–23340, 2022. 2

[70] Peng Wang, Shijie Wang, Junyang Lin, Shuai Bai, Xiaohuan Zhou, Jingren Zhou, Xinggang Wang, and Chang Zhou. One-peace: Exploring one general representation model toward unlimited modalities. *arXiv preprint arXiv:2305.11172*, 2023. 2

[71] Zhilin Xie, Kyungjae Lee, Howard Chen, Shuang Li, and Angel Chang. Refer360: Referring expression comprehension in 360° images. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[72] Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhengsu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. In *Proc. International Conference on Learning Representations (ICLR)*, 2024. 3

[73] Bin Yan, Yifan Jiang, Jiannan Wu, Dong Wang, Ping Luo, Zehuan Yuan, and Huchuan Lu. Universal instance perception as object discovery and retrieval. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7042–7052, 2023. 2

[74] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5525–5533, 2016. 2

[75] Zhongjie Yang, Boqing Gong, Liangliang Wang, Wei Huang, Dong Yu, and Jiebo Luo. A fast and accurate one-stage approach to visual grounding. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4683–4693, 2019. 1, 2

[76] Hantao Yao, Rui Zhang, and Changsheng Xu. Visual-language prompt tuning with knowledge-guided context optimization. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6767, 2023. 3

[77] Hantao Yao, Rui Zhang, and Changsheng Xu. Tcp: Textual-based class-aware prompt tuning for visual-language model. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[78] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Proc. European Conference on Computer Vision (ECCV)*, pages 69–85, 2016. 1, 2, 3

[79] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L Berg. A joint speaker-listener-reinforcer model for referring expressions. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7282–7290, 2017. 1, 2

[80] Licheng Yu, Zhe Lin, Xin Shen, Jimei Yang, Xiaohui Lu, Mohit Bansal, and Tamara L Berg. Mattnet: Modular attention network for referring expression comprehension. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1307–1315, 2018. 1, 2

[81] Xuehui Yu, Yuxuan Gong, Nianlong Jiang, Qixiang Ye, and Zhenjun Han. Scale match for tiny person detection. In *Proc. IEEE Conference on Winter Conference on Applications of Computer Vision (WACV)*, pages 1257–1265, 2020. 2

[82] Chunyuan Zhang, Junjun Li, Shoufa Chen, et al. Detrs with collaborative hybrid assignments training. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 1

[83] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4158–4166, 2018. 1, 2

[84] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2

[85] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: Detr with improved denoising anchor boxes for end-to-end object detection. In *Proc. International Conference on Learning Representations (ICLR)*, 2023. 1

[86] Ji Zhang, Shihan Wu, Lianli Gao, Hengtao Shen, and Jingkuan Song. Dept: Decoupled prompt tuning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[87] X. Zhang, Y. Li, G. Huang, Z. Wang, et al. An open and comprehensive pipeline for unified object grounding and detection. *arXiv preprint arXiv:2401.02361*, 2024. 2, 7

[88] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16816–16825, 2022. 3

[89] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2, 5, 7

[90] Beier Zhu, Yulei Niu, Yucheng Han, Yue Wu, and Hanwang Zhang. Prompt-aligned gradient for prompt tuning. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15659–15669, 2023. 3

[91] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Haibin Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(11):7380–7399, 2021. 2

[92] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *Proc. International Conference on Learning Representations (ICLR)*, 2020. 1

[93] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu. Traffic-sign detection and classification in the wild. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2110–2118, 2016. 2

[94] Bohan Zhuang, Qi Wu, Chunhua Shen, Ian Reid, and Anton van den Hengel. Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4252–4261, 2018. 1, 2