Referring Expression Comprehension for Small Objects

Supplementary Material

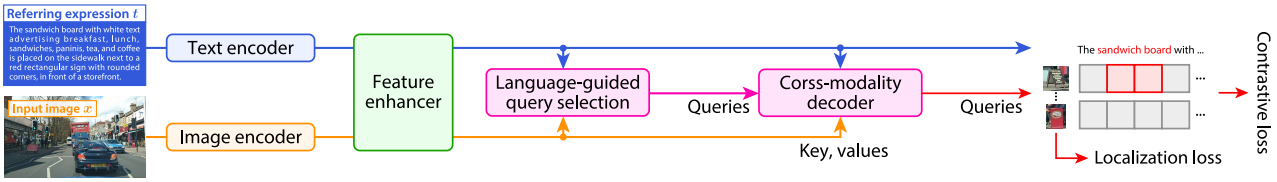


Figure 9. Architecture of GroundingDINO [42, 87].

A. Implementation details and analysis

Model architecture. Figure 9 shows the architecture of GroundingDINO [42], which we used as a backbone architecture in our experiments. It consists of five components: a text encoder, an image encoder, a feature enhancer, a language guided query selection module, and a cross-modality decoder. The BERT model is used as the text encoder. The Swin transformer is used as the image encoder. Figure 10 shows the architecture of the feature enhancer and the decoder, to which we applied parameter-efficient fine-tuning methods. Below, we describe details of each fine-tuning method.

Full fine-tuning. Full fine-tuning uses all parameters as learnable parameters. The number of parameters for each component is listed in Table 8.

PIZA-CoOp. CoOp is applied to the text encoder by prepending 16 learnable embeddings to input text prompt. PIZA-CoOp further inserts zooming-step embeddings $\boldsymbol{h}$ between the prepended learnable embeddings and the text prompt via learnable linear layers H. Specifically, H consists of L linear layers, $H_1, H_2, \cdots, H_L$, each of which is applied to $\boldsymbol{h}$ to obtain a sequence of embeddings whose length is $L = 8$. During fine-tuning, all LayerNorm layers are also updated. Ablation and hyperparameter studies are shown in Table 9. Although we also tried larger values for L, they did not lead to improved performance on the validation and Test-B sets. Overall, PIZA-CoOp did not surpass the results achieved by PIZA-LoRA and PIZA-Adapter+.

PIZA-LoRA. We applied PIZA-LoRA to self-attention and cross-attention layers in the feature enhancer and decoder. Figure 11 shows the detailed architecture. For the feature enhancer, PIZA-LoRA is applied to its text-to-image cross-attention, image-to-text cross-attention, and self-attention modules. For matrices to compute queries for the cross-attention modules, the zoom-step embedding is added to the LoRA bottleneck through a learnable matrix C. The vanilla LoRA is applied to the other liner functions in this module because we observed that inserting zoom-step em-

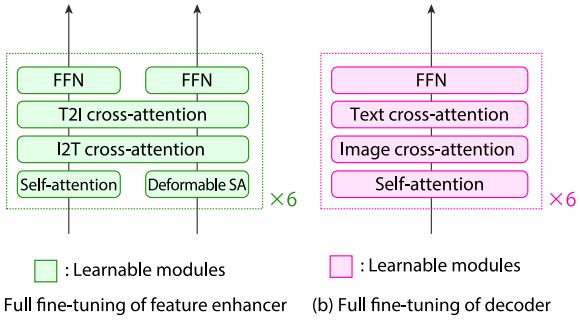


Figure 10. Block architectures of feature enhancer and decoder. Full fine-tuning updates all parameters.

Module	Architecture	#Params
Text encoder	BERT	108.9M
Image encoder	SwinT	27.5M
Feature enhancer	Figure 10 (a)	21.9M
Decoder	Figure 10 (b)	11.1M
Others	-	3.4M
Total	-	172.8M

Table 8. Number of parameters for each module.

Method	#Prm.	Val	Test-A	Test-B
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
$L = 4$	0.5M	26.4/39.2/29.3	29.5/41.0/34.2	21.9/33.9/24.4
$L = 8$	0.9M	26.3/39.1/29.7	29.4/41.2/34.2	21.9/33.8/24.3
$L = 16$	1.7M	25.8/38.0/29.1	29.8/41.5/34.5	21.1/32.5/23.5

Table 9. Ablation and hyperparameter studies for PIZA-CoOp. Train-S is used for training. Each triplet of values indicates mAcc/Acc_{50}/Acc_{75}.

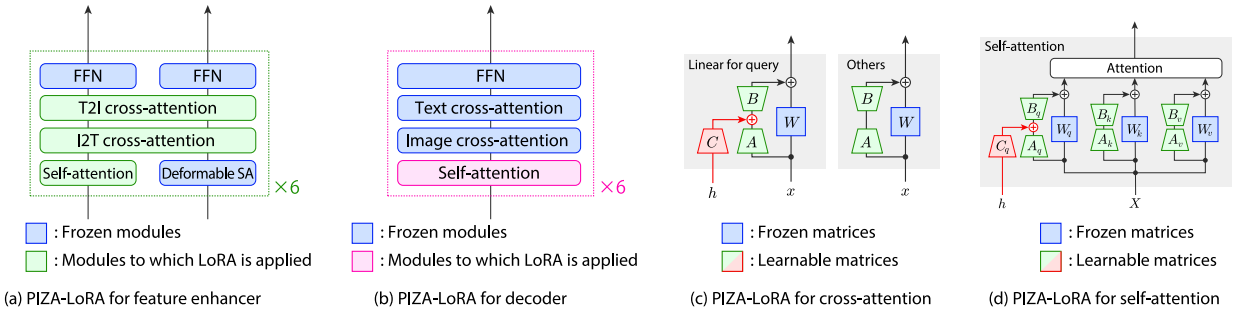


Figure 11. PIZA-LoRA architecture.

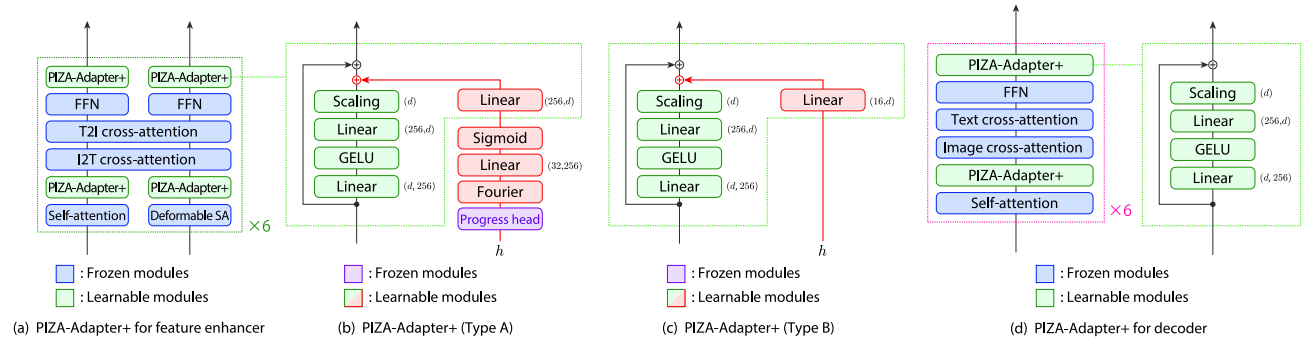


Figure 12. PIZA-Adapter+ architecture. Types A and B are designed for training with a small dataset and a large dataset, respectively.

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
$r = 64$	5.1M	31.1/44.9/35.2	34.4/47.3/40.0	25.7/38.5/28.9
$r = 16$	1.5M	30.9/44.7/34.9	33.8/46.6/39.2	25.8/38.7/28.9
$r = 4$	0.6M	30.8/44.6/34.7	33.7/46.4/39.1	25.8/38.5/29.0

Table 10. Ablation and hyperparameter studies for PIZA-LoRA. r indicates the rank of the low-rank matrices. Train-S is used for training. Each triplet of values indicate mAcc/Acc_{50}/Acc_{75}.

bedding did not improve the performance. For self-attention modules, we applied PIZA-LoRA in the same way. During fine-tuning, all LayerNorm layers are also updated. Ablation and hyperparameter studies are shown in Table 10. For PIZA-LoRA, increasing the rank to 64 slightly improved performance but did not achieve the performance level of PIZA-Adapter+ of Table 3.

PIZA-Adapter+. We applied PIZA-Adapter+ to the feature enhancer. As shown in Figure 12, four PIZA-Adapter+ modules are inserted into each feature enhancer block in a post-adapter manner, *i.e.*, adapters are inserted after the feedforward networks and attention modules. Each PIZA-Adapter+ module consists of either Type A in Figure 12 (b) or Type B in Figure 12 (c). Type A leverages the zoom progress value obtained from the frozen progress head of the PIZA module (the module to predict an EOS label with a progress value in Figure 5). The time embedding module that originates from the stable diffusion [61], consisting of a Fourier embedding and a small MLP, is then applied to the zoom progress value. Type B omits the progress head and uses the features extracted from the PIZA module. Comparison of Types A and B is shown in Table 11. Type A is particularly effective when a small size of dataset is used for training in experiments with Train-S. Although Type B is seemingly more efficient than Type A, it requires more training data than Type A because it lacks the progress head that enlarges the variety of conditioning inputs. We also designed PIZA-Adapter+ for the decoder in Figure 12 (d), but this did not improve the performance.

PIZA-VPT. VPT [26] is applied to the image encoder by prepending learnable embeddings to input visual prompt as $G(\boldsymbol{x}, \boldsymbol{t}) = F([\boldsymbol{e}, \boldsymbol{x}], \boldsymbol{t})$, where $\boldsymbol{e}$ is a sequence of learnable embeddings of length 16. PIZA-VPT inserts the zooming-step embeddings $\boldsymbol{h}$ as $G_{\circledast}(\boldsymbol{x}, \boldsymbol{t}) = F([\boldsymbol{e}, H(\boldsymbol{h}), \boldsymbol{x}], \boldsymbol{t})$, where H is the module consisting of L linear layers, similar to PIZA-CoOp. As shown in Table 12, PIZA-VPT outperformed VPT. Table 13 shows ablation and hyperparameter studies. Similar to CoOp, PIZA-VPT did not surpass the results achieved by PIZA-LoRA and PIZA-Adapter+.

Experiments in LMMs We conducted experiments with three large multimodal models (LMMs) that can perform

Method	#Prm.	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}
PIZA-Adapter+(Type A)	3.5M	36.8	53.5	41.8	43.1	59.6	50.1	30.4	45.9	34.1	37.0	59.2	40.2	43.1	64.7	48.9	29.7	50.3	30.9
PIZA-Adapter+(Type B)	3.5M	35.6	51.6	40.7	41.8	58.0	48.8	28.6	43.6	32.2	39.0	60.6	42.9	45.1	66.2	51.7	31.7	52.2	33.6

Table 11. Comparison of PIZA-Adapter+ configurations.

Method	#Prm.	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}
VPT	0.1M	19.8	35.3	19.9	23.9	39.2	25.5	15.4	29.6	14.3	22.5	39.8	22.6	27.2	44.5	29.3	17.4	33.5	16.4
PIZA-VPT (Ours)	0.6M	26.5	38.9	29.6	28.7	39.9	33.1	22.2	33.8	24.9	29.7	43.3	33.6	33.7	46.7	39.0	25.0	37.9	28.0

Table 12. Results for VPT and PIZA-VPT.

Method	#Prm.	Val	Test-A	Test-B
PIZA-VPT	0.6M	26.5/38.9/29.6	28.7/39.9/33.1	22.2/33.8/24.9
w/o emb. insertion	0.4M	25.9/37.9/29.4	28.2/39.0/32.7	21.8/32.8/24.3
w/o PIZA module	0.1M	19.8/35.3/19.9	23.9/39.2/25.5	15.4/29.6/14.3
$L = 4$	0.5M	25.7/37.3/28.9	27.9/38.7/32.6	21.7/32.7/24.2
$L = 8$	0.6M	26.5/38.9/29.6	28.7/39.9/33.1	22.2/33.8/24.9
$L = 16$	0.8M	26.0/38.0/29.6	27.8/38.5/32.1	21.7/32.8/24.3

Table 13. Ablation and hyperparameter studies for PIZA-VPT. Train-S is used for training. Each triplet of values indicates mAcc/Acc_{50}/Acc_{75}.

Method	#Prm.	Val	TestA	TestB
Zero-shot	0	50.4	57.2	43.2
Full fine-tuning	173M	89.2	91.9	86.0
Adapter+	3.5M	86.9	89.6	83.3
PIZA-Adapter+ (Ours)	3.5M	87.4	90.2	84.0

Table 14. Results on RefCOCO. Each value indicates Acc_{50}.

Method	LLM	Val	Test-A	Test-B
Zero-shot	Qwen2-VL-7B	0.2/0.8/0.0	0.3/1.1/0.1	0.1/0.3/0.0
Full FT	Qwen2-VL-7B	1.9/6.4/0.6	2.4/7.9/0.8	1.2/4.2/0.4
LoRA	Qwen2-VL-7B	3.8/12.4/1.4	5.0/15.5/2.0	2.6/8.7/0.9
PIZA-LoRA	Qwen2-VL-7B	27.9/49.4/28.4	31.8/52.4/34.5	23.0/42.6/22.2
Zero-shot	InternVL2.5-8B	0.0/0.0/0.0	0.0/0.1/0.0	0.0/0.0/0.0
Full FT	InternVL2.5-8B	0.1/0.5/0.0	0.1/0.5/0.0	0.1/0.5/0.0
LoRA	InternVL2.5-8B	0.2/0.8/0.0	0.2/0.8/0.1	0.2/0.8/0.0
PIZA-LoRA	InternVL2.5-8B	20.7/47.5/19.2	25.4/55.4/24.5	16.7/40.2/13.9
Zero-shot	LLaVA-NeXT-7B	0.0/0.0/0.0	0.0/0.0/0.0	0.0/0.0/0.0
Full FT	LLaVA-NeXT-7B	0.1/0.3/0.0	0.1/0.2/0.0	0.1/0.2/0.0
LoRA	LLaVA-NeXT-7B	0.7/2.7/0.1	0.7/2.5/0.1	0.6/2.2/0.2
PIZA-LoRA	LLaVA-NeXT-7B	10.8/27.7/6.3	12.0/30.4/7.2	8.7/23.1/5.0

Table 15. Experiments with LLMs on SOREC (Train-L). LoRA rank is set to 128. Each triplet of values indicates mAcc/Acc_{50}/Acc_{75}.

the REC task: Qwen2-VL-7B, InternVL-2.5-8B (n_{max}=24 for dynamic resolution), and LLaVA-NeXT-Mistral-7B. We utilized LoRA tuning with their prompts for REC. As shown in Table 15, our PIZA approach unlocks their abilities to perform REC on small objects and significantly boosts the performance. While full fine-tuning and LoRA tuning led to slight improvements, their performance was significantly lower compared to PIZA-LoRA.

RefCOCO. We also ran experiments on RefCOCO to demonstrate that methodological improvements with PIZA modules for small objects doesn't significantly impact the performance for objects of other sizes. RefCOCO is one of the first dataset for referring expression comprehension [30, 53, 57, 78]. Unlike the SOREC dataset for small objects, RefCOCO concentrates on referring expressions for objects that occupy a relatively large portion of the images in MS-COCO [39]. RefCOCO became a commonly-used benchmark of the referring expression comprehension task for a long time. As shown in Table 14, our PIZA-Adapter+ doesn't greatly decrease the accuracy on RefCOCO. PIZA-Adapter+ outperforms Adapter+ because the small learnable PIZA module helps improve the performance in RefCOCO. For larger objects, the `[EOS]` token was predicted after the first inference step.

Extended training dataset. The extended training dataset $\mathcal{E}$ consists of ground truth search processes P^*. The average length of P^* was 2.11, indicating that, in most cases, two zooming steps are sufficient to localize objects in the SOREC dataset via fine-tuning with PIZA. The hyperparameters λ_1, λ_2 of the weighting function were optimized for each target bounding box to ensure that the minimum size of the input image is larger than 450 pixels, as we empirically found that including smaller cropped images degrades the performance. Specifically, the parameters are first set to $\lambda_1 = 1.0$ and $\lambda_2 = 1.0$, and then if the edge

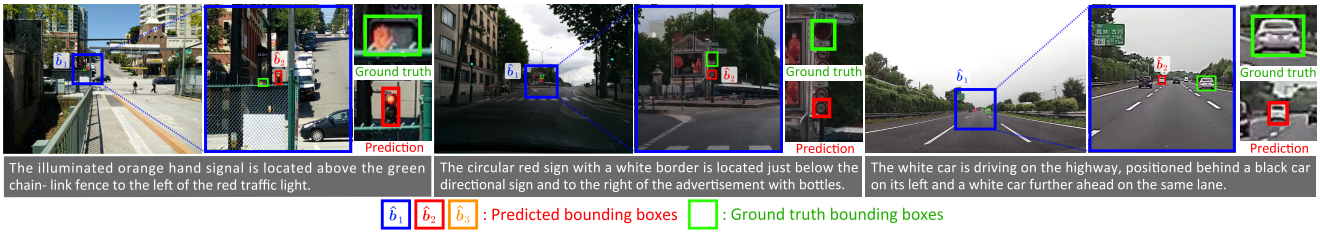


Figure 13. Failure cases.

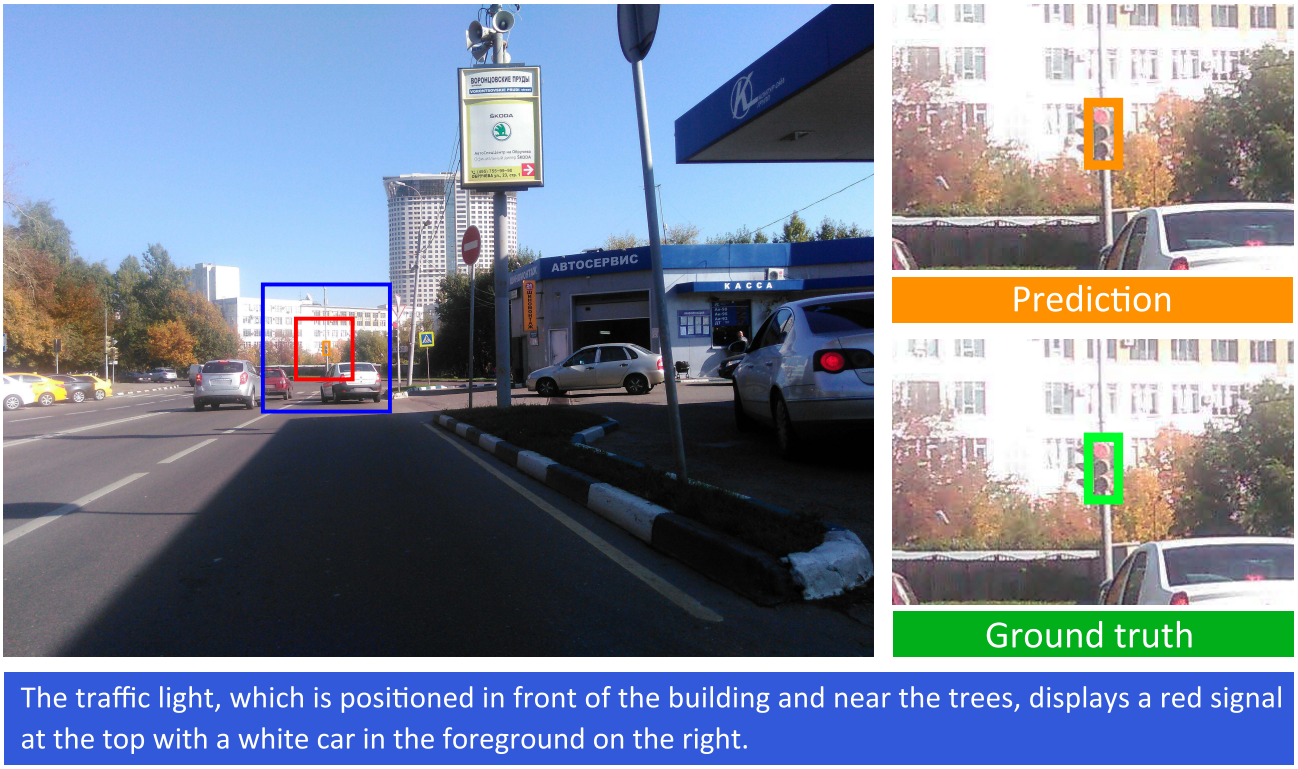


Figure 14. Qualitative examples.

length of the second-to-last bounding box is smaller than 450 pixels, we resample the search process by multiplying λ_2 by 1.1, iteratively until the edge length exceeds the threshold.

Error analysis. We analyzed failure cases, as shown in Figure 13. The results indicate that localizing objects occluded by other objects or placed in close proximity to similar objects remains challenging. Creating datasets with 8K or higher resolution images, which may require additional steps, is also left for future work.

Qualitative examples. Figures 14 to 19 show qualitative examples. The predicted bounding boxes, $\hat{\boldsymbol{b}}_1, \hat{\boldsymbol{b}}_2$, and $\hat{\boldsymbol{b}}_3$, are colored in blue, red, and orange, respectively, in each figure. The final predictions, $\hat{\boldsymbol{b}}_{\hat{T}}$, where $\hat{T} = 2$ or 3, are compared with the ground truth bounding boxes in green. As shown, our method successfully localizes extremely small target objects.

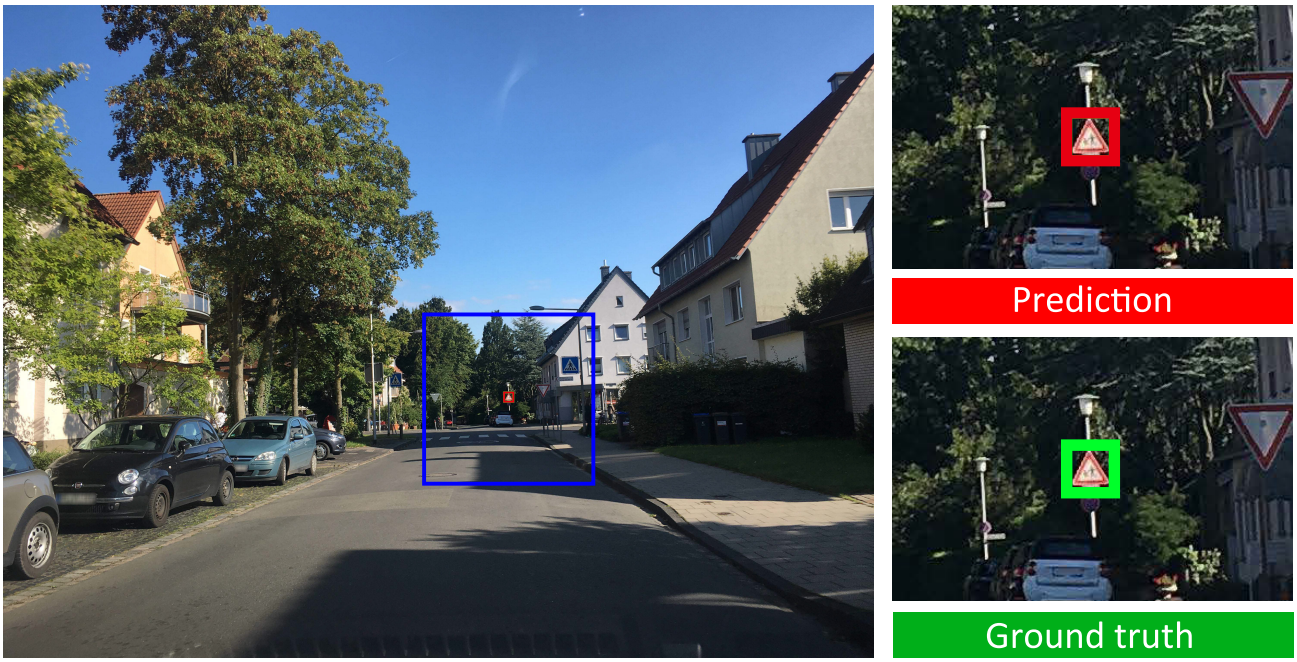
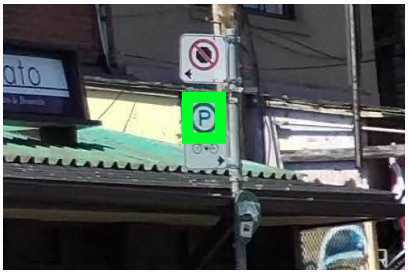


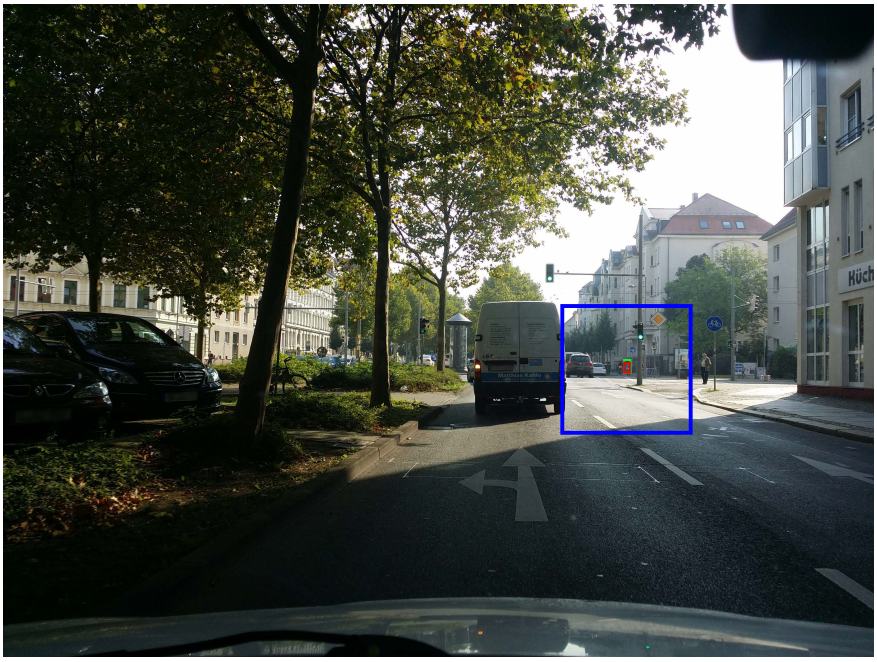
Figure 15. Qualitative examples.

Prediction



Ground truth

The sign with a blue "P" is located above the bicycle symbol and below the no parking sign mounted on a gray pole.



Prediction

Ground truth

The person wearing a gray outfit and riding a bicycle with a white seat is positioned between a person in a gray jacket and a person in a black jacket, near a street sign and a pole.

Figure 16. Qualitative examples.

Figure 17. Qualitative examples.

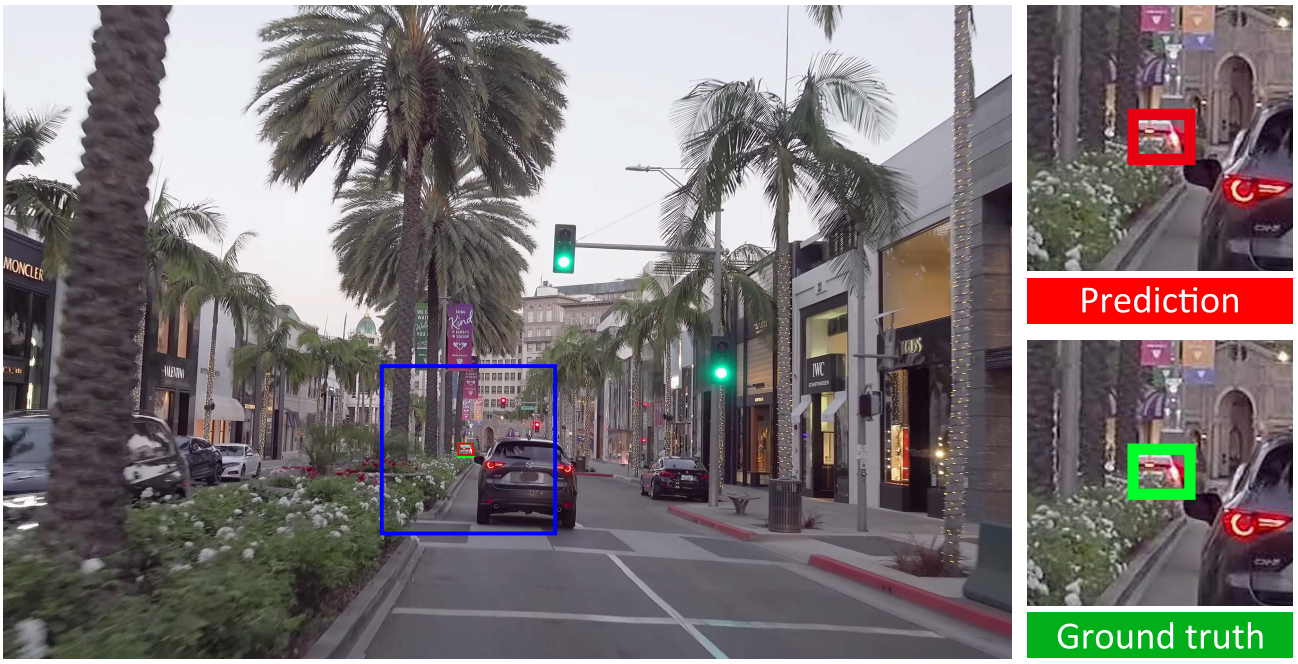
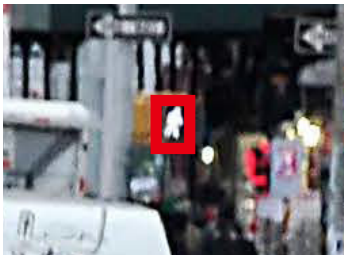
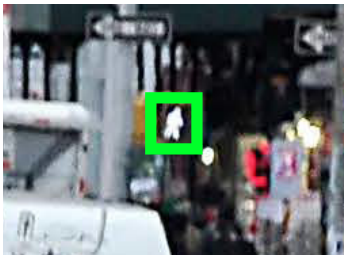


Figure 18. Qualitative examples.

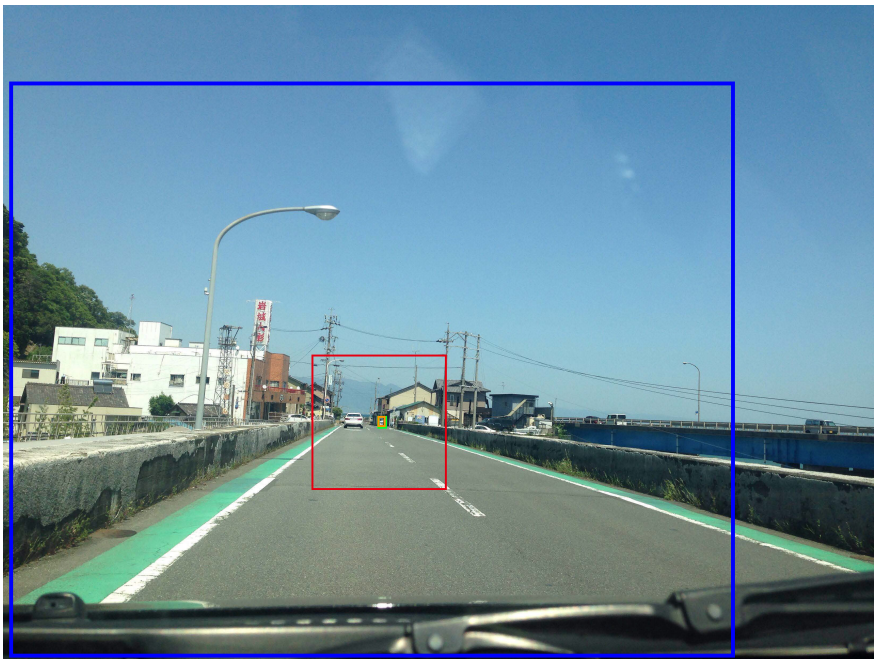


Prediction



Ground truth

The white pedestrian crossing signal is mounted on a yellow box, positioned above a blurry background with dark and light elements.



Prediction

Ground truth

The white truck is parked on the street between two buildings, with another truck visible further down the street with the backdrop of a mountain.

Figure 19. Qualitative examples.