# Exploring Contextual Attribute Density in Referring Expression Counting

Supplementary Material

## S1. Introduction

This supplementary material includes the following contents:

- Detailed network architectures and implementation of CAD-GD;
- Training and inference detail on FSC-147 and CARPK datasets;
- Parameters and training resource;
- Additional qualitative results on REC-8K;
- Limitations of CAD-GD.

## S2. Network Architectures

Here we detail some modules within our network.

### S2.1. Backbone and Feature Enhancer

The backbones consist of a visual backbone and a text backbone. The visual backbone is a Swin Transformer [6], we extract four different scales image features, which are $\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}$ of input width and height, with the visual backbone. Then we sent these image features into $1 \times 1$ convolutions to project into 256 dimensions. The text backbone is a BERT-based text transformer [3]. The text backbone maps the input referring expression $T$ to a sequence of at most 256 tokens. The encoded text tokens are 256 dimensional feature vectors. We feed the features from the visual backbone and the text backbone into the feature enhancer from GroundingDINO [5], composed of 6 blocks. Each block uses a deformable self-attention [8] to enhance the image features and a vanilla self-attention for text feature enhancement. Besides the self-attention mechanism, it contains an image-to-text cross-attention module and a text-to-image cross-attention module for feature fusion. After the feature enhancer, we can obtain the visual features $\{F_{vi}\}_{i=1}^{4}$ and text feature $F_t$.

### S2.2. U-shape CAD Estimator

Here we detail the architecture of the U-shape CAD Estimator. As Figure 1 shows, the U-shape CAD Estimator consists of layers of convolutions and bilinear upsampling. The intermediate features serve as CAD features.

### S2.3. Localization Decoder

The number of decoder queries is set to 900 as in GroundingDINO [5]. The localization decoder consists of 6 layers of decoder blocks. Each decoder block contains a self-attention layer for decoder queries, a visual cross-attention layer to combine CAD enhanced visual features, a text cross-attention layer to combine text features, and an FFN layer.

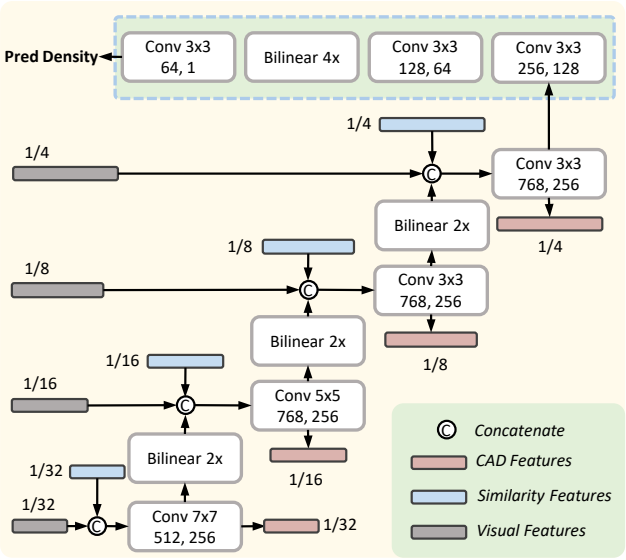


Figure 1. **U-shape CAD Estimator.** The visual features and the similarity features are sent into the U-shape CAD Estimator to obtain the CAD features. Each convolution block, in which the left and right dimensions are input and output dimensions separately, is followed by a ReLU as the activation function.

### S2.4. Localization Loss

The localization loss is the same as the loss in GroundingREC [2], which is used to optimize the location and classification of predicted points using ground truth points. By Bipartite Matching, we can first match the predicted points with the ground truth points, and then find the corresponding image tokens associated with the predicted points. The localization loss consists of a matching loss, a cross-entropy classification loss, and a contrastive loss.

**Matching loss.** The matching loss is for point regression which is the $L_1$ distance between the predicted points $\hat{p}_k$ and the ground truth points $p_k$. The total number of matched points is $K$. The $\mathcal{L}_{\text{match}}$ is calculated as follow:

$$\mathcal{L}_{\text{match}} = \frac{1}{K} \sum_{k=1}^{K} \|\hat{p}_k - p_k\|_1 . \tag{1}$$

**Cross-entropy classification loss.** Cross-entropy classification loss is for point classification, where $y_i$ is the ground truth label for $i^{th}$ text token and $\hat{y}_i$ is the class logit between $i^{th}$ text token and $k^{th}$ predicted image token. The

loss is calculated by taking the mean of all scores for $N$ text tokens and then averaging over $K$ matched points.

$$\mathcal{L}_{\text{cls}} = \frac{1}{K}\sum_{k=1}^{K}[-\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i^k + (1-y_i)\log(1-\hat{y}_i^k)]]. \quad (2)$$

**Contrastive loss.** Contrastive loss is for image-text alignment. For the REC-8K, we take multiple inputs of different attributes of the same class. For the $k^{th}$ matched image token, we take the corresponding attributes of the input referring expression as the positive text sample, and the mean of other attributes of the same class as the negative sample. Then we can obtain the positive score $s_k^p$ and negative similarity score $s_k^n$ by calculating the similarity of the $k^{th}$ matched image token with the positive sample and the negative sample separately. The final contrastive loss for all the matched image tokens is as follows:

$$\mathcal{L}_{\text{constrast}} = -\frac{1}{K}\sum_{k=1}^{K}[\log(s_k^p) + \log(1-s_k^n)]. \quad (3)$$

Note that in the FSC-147 [7] dataset, we take classes different from the corresponding class of the $k^{th}$ matched image token in a batch as negative samples, because FSC-147 only contains the classes without exact attributes.

The final localization function is as follows:

$$\mathcal{L}_{\text{loc}} = \mathcal{L}_{\text{match}} + \lambda_1 \mathcal{L}_{\text{cls}} + \lambda_2 \mathcal{L}_{\text{constrast}}, \quad (4)$$

where the $\lambda_1$ and $\lambda_2$ are set to 5 and 0.06 respectively.

### S2.5. Additional Implementation Detail

The mapping function in CAD Generate Module is a linear layer with 256 input features and 256 output features followed with a LayerNorm. The kernel size of the convolution layer in spatial attention is $7 \times 7$. The MLP in channel attention consists of 4 layers of linear with 256 input and output features. For the density init in CAD dynamic query module, we use a layer of cross-attention with a hidden dimension of 256 and 8 heads to fuse features. The density loss weight $\alpha$ is set to 1.

## S3. Training and Inference

Here we demonstrate the detailed training and inference setting for FSC-147 and CARPK datasets.

### S3.1. FSC-147 Dataset

**Training.** For the FSC-147 [7] dataset, CAD-GD is trained for 30 epochs using AdamW as the optimizer with a batch size of 4. The learning rate is set to $1e-5$ and decays by $10\times$ on the $15^{th}$ epoch. The visual backbone and text backbone are frozen during training. For the data augmentation, the minimum side length of the image is resized to a side length in {480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800} such that the aspect ratio of the image is maintained as in CountGD [1]. Following CountGD, all the classes in the FSC-147 training set are concatenated into a single caption with ".". The density maps for training are directly from the FSC-147 dataset.

**Inference.** At inference, each image is resized such that its shortest side length is 800 pixels, and its aspect ratio is maintained. The image is then normalized and passed to the model. We use the adaptive cropping strategy in CountGD to overcome the 900 counting quota of the model. In specific, when the prediction number is larger than 600, we then crop the image into 4 pieces without overlapping. Then we resize all these pieces with their shortest side length to 800 pixels. To obtain the final count, the number of detected instances in each crop window are added together. The threshold of FSC-147 is set to 0.3.

Table 1. **Comparison of the model size and FLOPs.** We obtain the FLOPs using a $384 \times 384$ image as input.

| Method | Backbone | Model Size (M) | GFLOPs |
|---|---|---|---|
| GroundingREC [2] | Swin-T | 144.1 | 67.6 |
| CAD-GD | Swin-T | 159.6 | 74.5 |
| GroundingREC [2] | Swin-B | 204.0 | 98.5 |
| CAD-GD | Swin-B | 219.5 | 105.4 |

### S3.2. CARPK Dataset

**Training.** We do not train CAD-GD on the CARPK [4] dataset to validate the cross-dataset generalization ability.

**Inference.** At inference, we do not use the resize and cropping strategy as in the FSC-147 dataset. For each image, we normalize it and then send it into the model for prediction. The threshold of CARKPK is set to 0.15.

## S4. Parameters and Training Resource

To verify the efficiency of our method, we compare our model size and floating point operations (FLOPs) with GroundingREC as shown in Table 1. The full training of CAD-GD with Swin-b visual backbone takes about 22GB on 1 Nvidia A100 GPU for about 20 hours.

## S5. Additional Qualitative Results

We provide additional qualitative results of our model as Figure 2 and Figure 3 show.

## S6. Limitations

Due to the limitation of query quota, it is difficult to count in a query image that contains more than 900 objects. Although we can use the cropping strategy to overcome the quota, it will introduce extra computational costs.

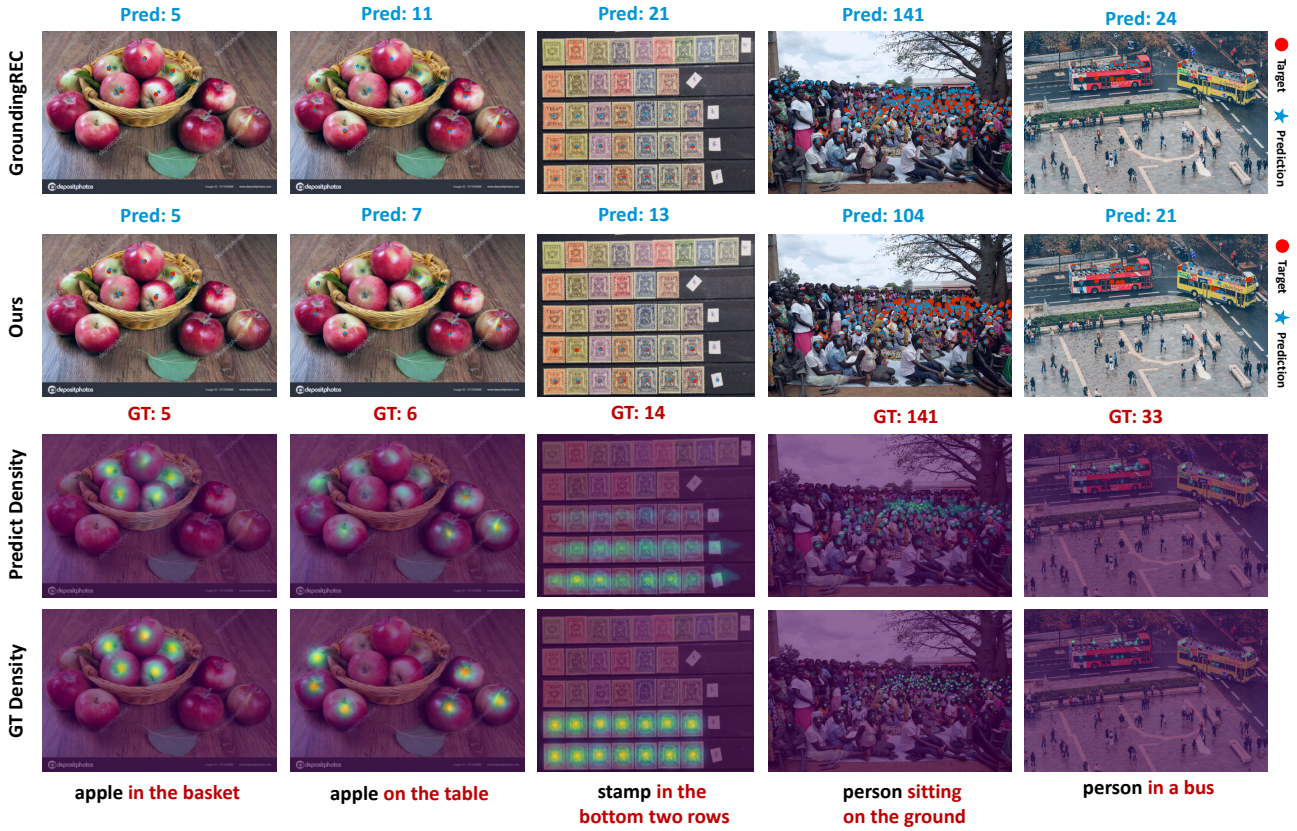


Figure 2. **Additional Qualitative Results 1.**

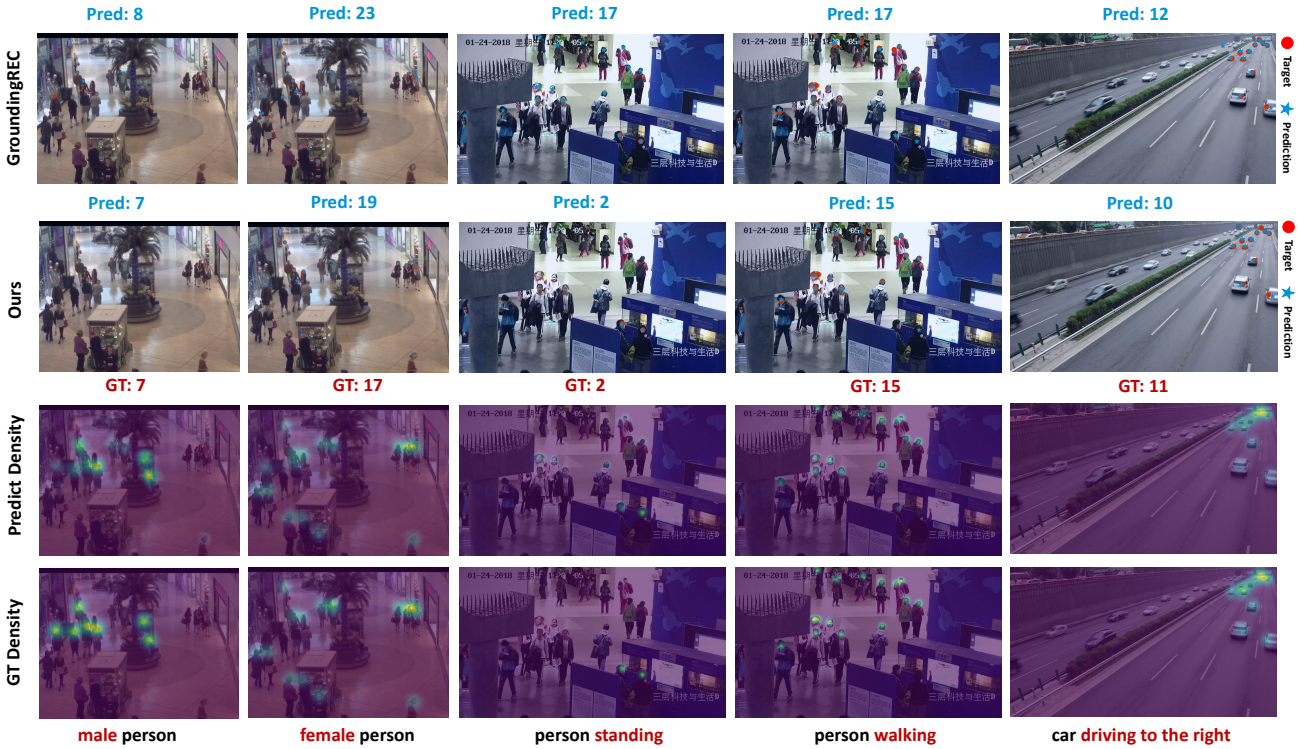


Figure 3. **Additional Qualitative Results 2.**