International Journal of Computer Vision (2022) 130:2501–2516
https://doi.org/10.1007/s11263-022-01647-z

Understanding Synonymous Referring Expressions via Contrastive Features

Yi-Wen Chen[1] · **Yi-Hsuan Tsai**[2] · **Ming-Hsuan Yang**[1]

Received: 12 September 2021 / Accepted: 29 June 2022 / Published online: 9 August 2022

Abstract
Referring expression comprehension aims to localize objects identified by natural language descriptions. This is a challenging task as it requires understanding of both visual and language domains. One nature is that each object can be described by synonymous sentences with paraphrases, and such varieties in languages have critical impact on learning a comprehension model. While prior work usually treats each sentence and attends it to an object separately, we focus on learning a referring expression comprehension model that considers the property in synonymous sentences. To this end, we develop an end-to-end trainable framework to learn contrastive features on the image and object instance levels, where features extracted from synonymous sentences to describe the same object should be closer to each other after mapping to the visual domain. We conduct extensive experiments to evaluate the proposed algorithm on several benchmark datasets, and demonstrate that our method performs favorably against the state-of-the-art approaches. Furthermore, since the varieties in expressions become larger across datasets when they describe objects in different ways, we present the cross-dataset and transfer learning settings to validate the ability of our learned transferable features.

Keywords Referring expression comprehension · Contrastive learning · Transfer learning · Synonymous sentences

1 Introduction

Referring expression comprehension is a task to localize a particular object within an image guided by a natural language description, e.g., "the man holding a remote standing next to a woman" or "the blue car". Since referring expressions are widely used in our daily conversations, the ability to understand such expressions provides an intuitive way for humans to interact with intelligent agents. One challenge of this task is to jointly comprehend the knowledge from both visual and language domains, where there are multiple ways (i.e., synonymous sentences) to describe and paraphrase the same object. In this paper, "synonymous sentences" mean different statements that describe the same object in an image. That is, synonymous sentences are various descriptions of the same object in a scene as annotators may use different words or adjectives. Synonymous sentences in this work are different from the general definition of "synonymous" that does not consider images. For instance, we can refer to an object by its attribute, location, or interaction with other objects. Referring expressions also vary in lengths and synonyms. As such, the varieties of sentences that describe the same object cause gaps in the language domain, and affect the model training process.

In this work, we take this language property, *synonymous sentences*, into consideration during the training process. This is different from existing referring expression comprehension methods (Liu et al., 2019; Yu et al., 2018; Yang et al., 2019a; Zhang et al., 2018) that do not explicitly consider synonymous sentences. Here, our main idea is to learn contrastive features when mapping the language features to the visual domain. That is, while the same object can be described by different synonymous sentences, these language features should be close to each other after mapping to the visual domain. On the other hand, for other expressions that do not

Communicated by Kwan-Yee Kenneth Wong.

✉ Ming-Hsuan Yang
mhyang@ucmerced.edu

Yi-Wen Chen
ychen319@ucmerced.edu

Yi-Hsuan Tsai
wasidennis@gmail.com

[1] University of California, Merced, CA, USA

[2] Phiar, Redwood City, CA, USA

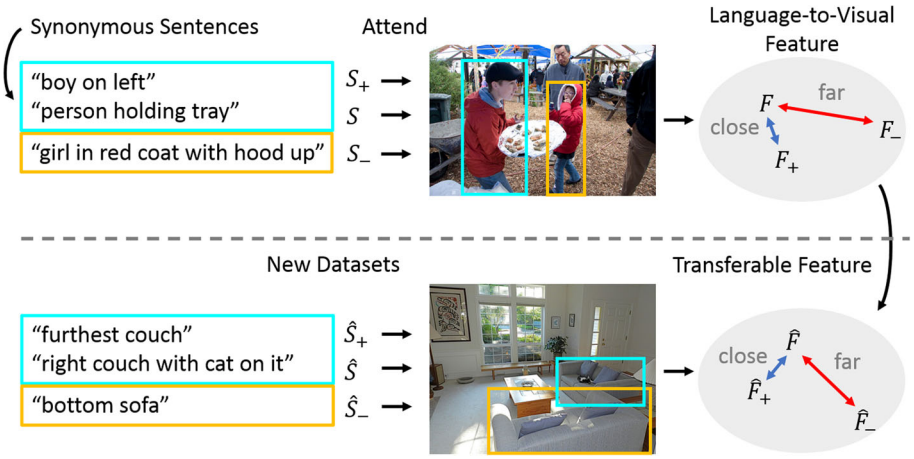

Fig. 1 Overview of the proposed algorithm. An object can be described in different ways, e.g., by its attribute, location, or interaction with other objects. For a referring expression S, there are positive expressions S_+ describing the same object and negative ones S_- for another object. While prior work considers each expression way separately, our method encourages features of synonymous sentences for the same object to attend nearby in the language-to-visual embedding space (F and F_+) but far away from negative ones (F_-). Thus, the proposed framework can transfer learned features to unseen data

describe that object, our model should also map them further away from that object (see Fig. 1 for an illustration).

To exploit how synonymous sentences are utilized to help model training as described above, we integrate feature learning techniques, e.g., contrastive learning (Hadsell et al., 2006; He et al., 2020; van den Oord et al., 2018), into our framework. Then, the requirement of (multiple) positive/negative samples in contrastive learning can be satisfied by the notion of synonymous sentences. However, it is not trivial to determine where we learn contrastive features in the model, in which we find that using language-to-visual features is beneficial to optimizing both the image and language modules. To this end, we design an end-to-end learnable framework that enables feature learning on two different levels with language-to-visual features, i.e., image and object instance levels, which are responsible for global context and relationships between object instances, respectively. Moreover, since there are large varieties of negative samples (i.e., any languages describing different objects can be negatives), we explore the option of mining negative samples to further facilitate the learning process.

In our framework, one benefit of learning contrastive features from synonymous sentences is to equip the model with the ability to contrast different language meanings. This ability is important when transferring the model to other datasets, as each domain may contain different varieties of sentences to describe objects. To understand whether the learned features are effectively transferred to a new domain, we show that our model performs better in the cross-dataset setting (testing on the unseen dataset), as well as in the transfer learning setting that fine-tunes our pre-trained model on the target dataset. Note that, although a similar concept of synonymous sentences is also adopted in (Wang et al., 2016) for retrieval tasks, it has not been exploited in referring expression comprehension for feature learning and transfer learning. Specifically, (Wang et al., 2016) uses off-the-shelf feature extractors and does not consider feature learning using multiple positive/negative samples on both image and instance levels like our framework.

We conduct extensive experiments on referring expression comprehension benchmarks to demonstrate the merits of learning contrastive features from synonymous sentences. First, we use the RefCOCO benchmarks, including RefCOCO (Yu et al., 2016), RefCOCO+ (Yu et al., 2016), and RefCOCOg (Mao et al., 2016; Nagaraja et al., 2016), to perform baseline studies with comparisons to state-of-the-art methods. Second, we focus on cross-dataset and transfer learning settings using the ReferItGame (Kazemzadeh et al., 2014) and Ref-Reasoning (Yang et al., 2020a) datasets to validate the ability of transferable features learned on the RefCOCO benchmarks.

The main contributions of this work are summarized as follows: (1) We propose a unified and end-to-end learnable framework for referring expression comprehension by considering various synonymous sentences to improve the training procedure. (2) We integrate feature learning techniques into our framework with a well-designed sampling strategy that learns contrastive features on both the image and instance levels. (3) We demonstrate that our model is able to effectively transfer learned representations in both cross-dataset and transfer learning settings.

2 Related Work

Referring Expression Comprehension The task of referring expression comprehension is typically considered as determining an object among several object proposals, given the referring expression. To this end, several methods adopt two-stage frameworks to first generate object proposals with a pre-trained object detection network, and then rank the proposals according to the expression. For example, CNN-LSTM models have been used to generate captions based on the image and proposals (Hu et al., 2016; Luo & Shakhnarovich, 2017; Mao et al., 2016), and the one with the maximum posterior probability for generating the query expression is selected. Other approaches (Rohrbach et al., 2016; Wang et al., 2016) embed proposals and the query sentence into a common feature space, and choose the object with the minimum distance to the expression. In addition, several strategies are adopted to improve the performance, e.g., analyzing the relationship between an object and its context (Nagaraja et al., 2016; Yu et al., 2016; Zhang et al., 2018), or exploring the attributes (Liu et al., 2017) to distinguish similar objects. To jointly consider multiple factors, MAttNet (Yu et al., 2018) learns a modular network by considering three components, i.e., subject appearance, location and relationship to other objects. While these two-stage frameworks achieve promising results, the computational cost is significantly high due to extensive post-processing steps and individual models. Furthermore, the model performance is largely limited by the pre-trained object detection network.

Some recent approaches adopt one-stage frameworks to tackle referring expression object segmentation (Chen et al., 2019), zero-shot grounding (Sadhu et al., 2019), and visual grounding (Huang et al., 2021; Yang et al., 2019b, 2020c, b), where the language features are fused with the object detector. In these methods, the models are end-to-end trainable and more computationally efficient. Compared to the methods mentioned above that consider each individual dataset separately, we aim to learn contrastive features by considering synonymous sentences during the training process. In this work, we adopt a one-stage framework in which the representations in both the language and visual domains are learned jointly.

Feature Learning Feature learning aims to represent data in the embedding space, where similar data points are close to each other, and dissimilar ones are far apart, based on pairwise (Sohn, 2016), triplet (Schroff et al., 2015) or contrastive relationships (Hadsell et al., 2006; He et al., 2020; van den Oord et al., 2018). This representation model is then used for downstream tasks such as classification and detection. These loss functions are computed on anchor, positive and negative samples, where the anchor-positive distance is minimized, and the anchor-negative distance is maximized. While the triplet loss (Schroff et al., 2015) uses one positive and one negative sample per anchor, contrastive loss (Hadsell et al., 2006; He et al., 2020; van den Oord et al., 2018) includes multiple positive and negative samples for each anchor, which makes the learning process more efficient.

For natural language processing tasks, recent studies (Devlin et al., 2018; Peters et al., 2018) based on the transformer (Vaswani et al., 2017) have shown success in transfer learning. Built upon the transformer-based BERT (Devlin et al., 2018) model, learning representation for vision and language tasks by large-scale pre-training methods recently attracts much attention. These methods (Chen et al., 2020b; Gan et al., 2020; Lu et al., 2019, 2020; Li et al., 2020; Zhou et al., 2020) learn generic representations from a large amount of image-text pairs in a self-supervised manner, and fine-tune the model for the downstream vision and language tasks. Recently, ViLBERT (Lu et al., 2019) and its multi-tasking version (Lu et al., 2020) use two parallel BERT-style models to extract features on image regions and text segments, and connect the two streams with co-attentional transformer layers. Moreover, the OSCAR (Li et al., 2020) method uses object tags as anchor points to align the vision and language modalities in a shared semantic space. While these approaches aim to learn generic representations for vision and language tasks by training the models on large-scale datasets, we focus on the referring expression comprehension, and adopt the feature learning techniques to improve the performance by considering synonymous sentences.

In this work, with a similar spirit to feature learning, we integrate the contrastive loss into our model by considering synonymous sentences for referring expression comprehension. While it is natural to use the concept of synonymous expressions in tasks such as image-text retrieval (Wang et al., 2016) or text-based person retrieval (Yamaguchi et al., 2017), it has not been exploited in referring expression comprehension to improve feature learning and further for transfer learning. Different from (Wang et al., 2016; Yamaguchi et al., 2017) that apply hinge loss on triplets, we consider multiple positive and negative samples for each anchor, and perform contrastive learning on both image and instance levels. Moreover, we adopt an end-to-end framework, where the visual and language features are jointly learned, which is different from (Wang et al., 2016; Yamaguchi et al., 2017) that use off-the-shelf feature extractors. Compared to MAttNet (Yu et al., 2018), our method leverages synonymous sentences via learning contrastive features, which has not been exploited in prior art. In addition, we apply contrastive loss differently in terms of sample construction, feature space and negative mining strategy. It is also worth mentioning that we apply the loss on image-level and instance-level features that are mapped from the language domain to the image domain in an end-to-end learning manner. With the proposed feature learning techniques, we demonstrate that our model is able

to transfer learned representations in both cross-dataset and transfer learning settings.

3 Proposed Framework

In this work, we address the problem of referring expression comprehension via using the information in *synonymous sentences* to learn contrastive features across sentences. Given an input image I and a referring expression $S = \{w_t\}_{t=1}^{T}$ consisting of T words w_t, the task is to localize the object identified by the expression. We design a framework composed of a visual encoder E_v and a language encoder E_l for feature extraction in the visual and language domains. Since our goal is to learn contrastive features after mapping the language features to the visual domain, we utilize the attention modules A_{img} and A_{ins} for language-to-visual features and a graph convolutional network (GCN) G for aggregating instance-level features, which will be detailed in the following sections. The output of referring expression comprehension is predicted by a detection head D. Fig. 2 shows the pipeline of the proposed framework.

Our method learns contrastive features on two levels to account for both the global context and relationships between object instances. To this end, we enforce that the language-to-visual features on either the image or instance level should be close to each other if the referring expressions are synonymous sentences, and vice versa. Given the image-level attention map $R_{l\to v}$ obtained from A_{img} and the instance-level attention feature $H_{l\to v}$ inferred from A_{ins} followed by a GCN module G, we regularize $R_{l\to v}$ and $H_{l\to v}$ by leveraging feature learning techniques, guided by the notion of synonymous sentences. As a result, our language-to-visual features contrast the attentions from the language domain to the visual one based on the sentence meanings, which facilitates the comprehension task.

3.1 Image-Level Feature Learning

To comprehend the information from the input image and referring expression, features of the two inputs are first extracted by each individual encoder. We then use an attention module A_{img} that attends the l-dimensional language feature $F_l = E_l(S) \in \mathbb{R}^l$ to the v-dimensional visual feature $F_v = E_v(I) \in \mathbb{R}^{h\times w\times v}$, where h and w are the spatial dimensions. A response map $R_{l\to v}$ that contains the multimodal information is obtained accordingly, i.e., $R_{l\to v} = A_{img}(F_l, F_v) \in \mathbb{R}^{h\times w}$. The details of the encoders and attention module A_{img} are presented later in Sect. 3.3.

As there are numerous synonymous sentences to describe the same object, the language-to-visual features should attend to similar regions regardless of how to describe the object. Intuitively, we can apply a triplet loss on the response map to encourage the samples with synonymous expressions describing the same object to be mapped closely in the embedding space. Otherwise, they should be mapped far away from each other.

Specifically, for each input image I and a referring expression anchor S, we randomly sample a positive expression S_+ that describes the same object, and a negative expression S_- identifying a different object within the same image. The triplet loss is then computed on the response generated by

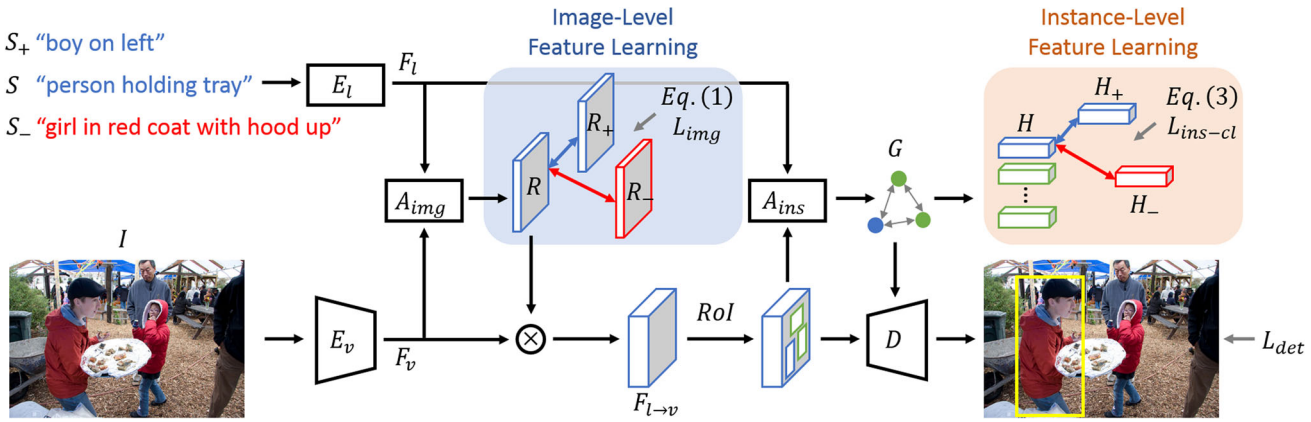

Fig. 2 Pipeline of the proposed framework. The features of the input image I and referring expression S (with its synonymous sentence S_+ and a negative expression S_-) are first extracted by a visual encoder E_v and language encoder E_l, respectively. Then we adopt two attention modules A_{img} and A_{ins} for attending language features to the visual domain on the image and instance levels, where we apply our feature learning losses L_{img} and L_{ins-cl} to contrast positive/negative pairs, i.e., $\{R_+, R_-\}$ and $\{H_+, H_-\}$ on two levels, respectively. On the instance level, a graph convolutional network G is employed to model the relationships between object proposals. Finally, we generate the object bounding box with a detection head D

attending the three expression samples to the image I:

$$L_{img} = \max(d(R, R_+) - d(R, R_-) + \alpha, 0), \quad (1)$$

where R, R_+ and R_- are the responses of the anchor, positive and negative samples, respectively. In addition, d is the L2 distance between two response maps and α is the margin. After this step, we combine $R_{l\to v}$ and F_v via element-wise multiplication $\otimes$ to produce the attentive feature $F_{l\to v} = R_{l\to v} \otimes F_v \in \mathbb{R}^{h\times w\times v}$, which is then used as the input to the detection head D (see Fig. 2). We note that the triplet loss is applied to the response map $R_{l\to v}$ rather than the feature $F_{l\to v}$, since $R_{l\to v}$ is easier to optimize with the much lower dimension.

3.2 Instance-Level Feature Learning

In addition to applying the triplet loss on the image level for learning contrastive features, we also consider the features on the instance level to encourage the model to focus more on local cues. However, through the RoI module that generates proposals, each proposal (instance) only contains the information within the receptive field of its bounding box but does not provide the local context information (e.g., interactions with other objects) described in the referring expression. To tackle this problem, we design a graph convolutional network (GCN) (Kipf & Welling, 2017) in a way similar to DGA (Yang et al., 2019a) to model the relationships between proposals, and then use the features after GCN as the input to our instance-level feature learning module. Different from DGA that applies pre-trained features to the GCN, we integrate the GCN in our end-to-end model. More details regarding the implementation are provided in Sect. 3.3.

Contrastive Feature Learning As shown in Fig. 2, similar to the image-level in Sect. 3.1, we first adopt an instance-level attention module A_{ins} following (Yu et al., 2018) to attend the language feature F_l to the RoI proposals, followed by the GCN module G to aggregate the proposal relationships. As a result, we obtain the instance-level language-to-visual features $H_{l\to v} = G(A_{ins}(F_l, \text{RoI}(F_{l\to v})))$.

Next, we propose to regularize $H_{l\to v}$ guided by the concept of synonymous sentences, where the proposal features from referring expressions that describe the same object should be close to each other. Otherwise, they should be apart from each other. To this end, one straightforward way to learn instance-level contrastive features is to apply the triplet loss similar to (1):

$$L_{ins-tri} = \max(d(H, H_+) - d(H, H_-) + \alpha, 0), \quad (2)$$

where H, H_+, and H_- represent instance-level features of the anchor, positive and negative expressions attending on the visual domain, respectively. Note that each expression may generate different RoI locations. To use the same proposals across samples in the triplet, we select the proposal with the highest IoU score with respect to the ground truth bounding box.

Contrastive Loss with Negative Mining Although the triplet loss in (2) can be used to learn instance-level contrastive features, it is limited to sample one positive and negative at a time, which may not fully exploit multiple synonymous sentences. Therefore, we leverage the contrastive loss (Khosla et al., 2020) with the property that can consider multiple positive/negative samples. Intuitively, we can treat synonymous sentences as positives, but the space of negative samples is large and noisy as any two languages describing different objects are negatives. Moreover, it has been studied that finding good negative samples is critical for learning contrastive features effectively (Kalantidis et al., 2020).

To tackle this issue, we employ the following two strategies to mine useful negative samples: (1) From the perspective of the visual domain, we mine samples that describe the same object category but in different images, and then use the corresponding referring expressions as the negatives. This encourages our model to contrast features that describe similar contents across images, as sentences referring to the same object category usually share common contexts. (2) Considering the language embedding features, we mine the top N samples (i.e., $N = 8$ in this work) that have closer language features to the anchor sample but in different images. This helps the model contrast samples that have a similar language structure. Overall, our contrastive loss L_{ins-cl} can be formulated as:

$$-\log \frac{\sum_{H_+\in Q_+} e^{h(H)^\top h(H_+)/\tau}}{\sum_{H_+\in Q_+} e^{h(H)^\top h(H_+)/\tau} + \sum_{H_-\in Q_-} e^{h(H)^\top h(H_-)/\tau}}, \quad (3)$$

where Q_+ and Q_- are the sets of positive and negative samples, and τ is the temperature parameter. In practice, we follow the SimCLR (Chen et al., 2020a) method and use $h(\cdot)$ as a linear layer that projects features H to an embedding space where the contrastive loss is applied.

Discussions Compared to the instance-level triplet loss in (2), using the contrastive loss in (3) has a few merits. First, given an anchor sample, it can contrast with multiple positive and negative samples at the same time, which is much more efficient than sampling the triplets, as shown in (Khosla et al., 2020) and our ablation study presented later. Second, the projection head $h(\cdot)$ provides a learnable buffer before feeding features to compute the contrastive loss, which helps the model learn better representations.

In terms of the sampling strategy, different from the negative mining method in (Chen et al., 2020c) that generates negatives with the same object category, we also consider

negatives with similar languages to the anchor language but containing other object categories. This difference allows us to sample more negatives with similar language structures, which is crucial for our contrastive loss in language.

3.3 Model Training and Implementation Details

In this section, we provide more details in training our framework and the design choices.

Overall Objective The overall objective for the proposed algorithm consists of the aforementioned loss functions (1) and (3) for learning contrastive features on the image and instance levels, and the detection loss L_{det} as defined in the Mask R-CNN (He et al., 2017) following the MAttNet (Yu et al., 2018) method:

$$L_{all} = L_{det} + L_{img} + L_{ins-cl}. \tag{4}$$

Implementation Details For the visual encoder E_v in our framework and the detection head D, we adopt the Mask R-CNN (He et al., 2017) as the backbone model, which is pre-trained on COCO training images, excluding those in validation and testing splits of RefCOCO, RefCOCO+ and RefCOCOg. The ResNet-101 (He et al., 2016) is used as the feature extractor, where the output of the final convolutional layer in the fourth block is the feature F_v that serves as the input to the attention module A_{img}. For the language encoder E_l, we use either the BERT (Devlin et al., 2018) or Bi-LSTM model. In the image-level attention module A_{img}, we adopt dynamic filters similar to (Chen et al., 2019) to attend language features to the visual domain and generate $R_{l\to v}$.

For the GCN in our method, the object proposals are generated from Mask R-CNN (He et al., 2017). We keep the top K detection candidates for each image, where K is set to 20. We then construct a graph $G = (\mathcal{V}, \mathcal{E})$ from the set of object proposals $P = \{p_i\}_{i=1}^{K}$, where each vertex $v_i \in \mathcal{V}$ corresponds to an object proposal p_i, and each edge $e_{ij} \in \mathcal{E}$ models the pairwise relationship between instances p_i and p_j. In the instance-level attention module A_{ins}, we compute the word attention on each object proposal to focus on instances that are referred to by the sentence. The word attention a_i on the proposal p_i is defined as the average of all the probabilities that each word w_t refers to p_i: $a_i = \frac{1}{T}\sum_{t=1}^{T} s_{i,t} = \frac{1}{T}\sum_{t=1}^{T}\langle f_{w_t}, f_{p_i}\rangle$, where T is the number of words in the sentence, and $s_{i,t}$ is the inner product between the feature f_{w_t} of word w_t and the average pooled feature f_{p_i} of proposal p_i. To compute the feature node f_{v_i} at vertex v_i, we first concatenate the average pooled feature f_{p_i} and the 5-dimensional location feature (Mao et al., 2016) $[\frac{x_{tl}}{W}, \frac{y_{tl}}{H}, \frac{x_{br}}{W}, \frac{y_{br}}{H}, \frac{wh}{WH}]$ on proposal p_i, where (x_{tl}, y_{tl}) and (x_{br}, y_{br}) are the coordinates of the top-left and bottom-right corners of the proposal, h and w are height and width of the proposal, H and W are height and width of the image. Then, the concatenated feature is multiplied by the word attention a_i to form f_{v_i}. We use a two-layer GCN to capture second-order interactions. Features after GCN are duplicated to each spatial location and concatenated with the spatial features after RoI. Then, the concatenated features are fed to the detection head D to generate final results. During testing, the detected object with the largest score is considered as the prediction. The margin α in the triplet loss (1) and (2) is set to 1. In the contrastive loss (3), the temperature τ is set to 0.1, and the projection head $h(\cdot)$ is a 2-layer MLP, projecting the features to a 128-dimensional latent space.

We implement the proposed model in PyTorch with the SGD optimizer, and the entire model is trained end-to-end with 10 epochs. The batch size is set to 8. The initial learning rate is set to 10^{-4} and decreased to 10^{-5} after 3 epochs. The total training time of our model is about 60 hours. The inference time is 0.325 seconds per frame. Our framework is implemented on a machine with an Intel Xeon 2.3 GHz processor and an NVIDIA GTX 1080 Ti GPU with 11 GB of memory. We implement our baseline as the same architecture without using contrastive learning, i.e., only L_{det} as the objective.

4 Experimental Results

We evaluate the proposed framework on three referring expression datasets, including RefCOCO (Yu et al., 2016), RefCOCO+ (Yu et al., 2016) and RefCOCOg (Mao et al., 2016; Nagaraja et al., 2016). The three datasets are collected on the MSCOCO (Lin et al., 2014) images, but with different ways to generate referring expressions. Extensive experiments are conducted in multiple settings. We first compare the performance of the proposed algorithm with state-of-the-art methods and present the ablation study to show the improvement made by each component. Then we evaluate the models on the unseen Ref-Reasoning (Yang et al., 2020a) dataset to validate the effectiveness on unseen datasets. Furthermore, we conduct experiments in the transfer learning setting, where the pre-trained models are fine-tuned on either the Ref-Reasoning (Yang et al., 2020a) or ReferItGame (Kazemzadeh et al., 2014) dataset. The source code and trained models will be made available to the public.

Intra- and Inter-Dataset Feature Learning Since synonymous sentences exist within one dataset and across datasets[1], we consider both intra- and inter-dataset feature learning loss. For each input image and referring expression anchor, we sample positive and negative expressions from the same dataset for the intra-dataset case, and expressions from dif-

[1] RefCOCO, RefCOCO+, and RefCOCOg datasets have the same images but with different ways to describe the same object using synonymous sentences.

ferent datasets as inter-dataset samples. In our experiments, we use the intra-dataset loss for training on a single dataset, and both the intra- and inter-dataset losses for jointly training on the three datasets.

4.1 Datasets and Evaluation Metric

RefCOCO contains 19,994 images with 142,209 referring expressions for 50,000 objects, while **RefCOCO+** is composed of 141,564 expressions for 49,856 objects in 19,992 images. Restrictions are not placed on generating expressions for RefCOCO, but put on RefCOCO+ by forbidding the location information, making it focus more on the appearance of the target object and its interaction with others. The two testing splits testA and testB are generated respectively on images containing multiple people and images containing multiple objects of other categories. We follow the split of the training, validation and testing images in (Yu et al., 2016), and there is no overlap across the three sets.

The **RefCOCOg** dataset consists of 85,474 referring expressions for 54,822 objects in 26,711 images with longer expressions. There are two splits constructed in different ways. The first split (Mao et al., 2016) randomly partitions objects into training and validation sets. Therefore, the same image could appear in both sets. The validation set is denoted as "val∗" in this paper. The second partition (Nagaraja et al., 2016) randomly splits images into training, validation and testing sets, where we denote the validation and testing splits as "val" and "test", respectively. In our experiments, when jointly training on three datasets, to avoid overlaps between the training, validation and testing images, we create another split for RefCOCOg, where each set contains the images present in the corresponding set of RefCOCO and RefCOCO+. We denote this split as "RefCOCOg∗".

The **Ref-Reasoning** dataset consists of 83,989 images from the GQA set (Hudson & Manning, 2019) with 791,956 referring expressions automatically generated based on scene graphs. The **ReferItGame** dataset is collected in an interactive game interface with images from the ImageCLEF set (Escalante et al., 2010). It contains 130,525 expressions, referring to 96,654 objects in 19,894 images. To evaluate the detection performance, the predicted bounding box is considered correct if the intersection-over-union (IoU) of the prediction and the ground truth bounding box is above 0.5.

4.2 Evaluation on Seen Datasets

Table 1 shows the results of the proposed algorithm against state-of-the-art methods (Chen et al., 2021; Deng et al., 2021; Huang et al., 2021; Luo & Shakhnarovich, 2017; Liu et al., 2017, 2019; Liao et al., 2020; Nagaraja et al., 2016; Yu et al., 2018; Yang et al., 2019a, b, 2020c; Yu et al., 2017; Zhuang et al., 2018; Zhang et al., 2018). All the compared approaches except for the recent methods (Deng et al., 2021; Huang et al., 2021; Liao et al., 2020; Yang et al., 2019b, 2020c) adopt two-stage frameworks, where the prediction is chosen from a set of proposals. Therefore, their models are not end-to-end trainable, while our one-stage framework is able to learn better feature representations by end-to-end training. Moreover, all of these methods train on each dataset separately and do not consider the varieties of synonymous sentences within/across datasets.

The models in the top and middle groups of Table 1 are trained and evaluated on the same single dataset. We separate the methods using different language encoders for fair comparisons. The results show that the full model trained with the proposed loss functions consistently improves our baseline model without considering synonymous sentences. In addition, our method with either LSTM or BERT as the language encoder performs favorably against most existing approaches. Our method also provides better performance than the transformer-based TransVG (Deng et al., 2021) method on 6 out of 9 splits. While Ref-NMS (Chen et al., 2021) outperforms the proposed method, they focus on improving two-stage methods. Therefore, the applications of Ref-NMS are limited.

For the runtime comparison, our unified framework (0.325 s per frame) is much faster than the two-stage MAttNet (Yu et al., 2018) and CM-Att-Erase (Liu et al., 2019) methods (0.671 and 0.734 s, respectively). While we use the same backbone (Mask R-CNN with ResNet-101) as MAttNet, our end-to-end model is more efficient by processing each image with a single stage. In contrast, MAttNet requires multiple steps of inference (i.e., 0.302 s for object bounding box generation, 0.276 s for region feature extraction, and 0.093 s for prediction).

The bottom group of Table 1 shows the results of our models jointly trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ (our split) datasets. Since the three datasets share the same images but contain expressions of very different properties, directly training on all of them as the baseline would cause training difficulties in a single model due to the large varieties in language. In contrast, by applying the proposed loss terms, the performance improves from the baseline, and the gains are larger than those in the single dataset setting. We also note that all the training images are the same, but the varieties of synonymous sentences are different across these two settings.

4.3 Ablation Study

We present the ablation study results in Table 2 to show the effect of each component in the proposed framework. The models are jointly trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets. In the top group of Table 2, we first demonstrate that applying either the image-level or instance-

Table 1 Comparisons with state-of-the-art methods

Method	Language Encoder	RefCOCO			RefCOCO+			RefCOCOg		
		val	testA	testB	val	testA	testB	val∗	val	test
Nagaraja et al. (2016)	LSTM	57.30	58.60	56.40	–	–	–	–	–	49.50
Luo and Shakhnarovich (2017)	LSTM	–	67.94	55.18	–	57.05	43.33	49.07	–	–
SLR (Yu et al., 2017)	LSTM	69.48	73.71	64.96	55.71	60.74	48.80	–	60.21	59.63
Liu et al. (2017)	LSTM	–	72.08	57.29	–	57.97	46.20	52.35	–	–
PLAN (Zhuang et al., 2018)	LSTM	–	75.31	65.52	–	61.34	50.86	58.03	–	–
VC (Zhang et al., 2018)	LSTM	–	73.33	67.44	–	58.40	53.18	62.30	–	–
MAttNet (Yu et al., 2018)	LSTM	76.65	81.14	69.99	65.33	71.62	56.02	–	66.58	67.27
CM-Att-Erase (Liu et al., 2019)	LSTM	78.35	83.14	71.32	68.09	73.65	58.03	–	67.99	68.67
DGA (Yang et al., 2019a)	LSTM	–	78.42	65.53	–	69.07	51.99	–	–	63.28
Darknet-LSTM (Yang et al., 2019b)	LSTM	73.66	75.78	71.32	–	–	–	–	–	–
RCCF (Liao et al., 2020)	LSTM	–	81.06	71.85	–	70.35	56.32	–	–	65.73
Ref-NMS (Chen et al., 2021)	GRU	80.70	84.00	76.04	68.25	73.68	59.42	–	70.55	70.62
LBYL-Net (Huang et al., 2021)	LSTM	78.76	82.18	71.91	66.67	73.21	56.23	58.72	–	–
Ours (baseline)	LSTM	76.63	79.59	69.93	63.67	71.26	55.04	61.71	66.24	65.85
Ours (full model)	LSTM	79.09	82.86	72.64	66.81	74.23	58.08	64.80	69.14	68.86
Darknet-BERT (Yang et al., 2019b)	BERT	72.05	74.81	67.59	55.72	60.37	48.54	48.14	59.03	58.70
ReSC-Large (Yang et al., 2020c)	BERT	77.63	80.45	72.30	63.59	68.36	56.81	63.12	67.30	67.20
LBYL-Net (Huang et al., 2021)	BERT	79.67	82.91	74.15	68.64	73.38	59.49	62.70	–	–
TransVG (Deng et al., 2021)	BERT	81.02	82.72	78.35	64.82	70.70	56.94	67.02	68.67	67.73
Ours (baseline)	BERT	77.02	80.25	70.53	64.31	71.83	55.20	62.27	66.48	66.34
Ours (full model)	BERT	79.63	83.32	73.27	67.49	74.85	58.42	65.23	69.67	69.51
Ours (baseline-all)	BERT	75.17	79.53	68.72	63.42	70.55	53.38	–	–	–
Ours (full model-all)	BERT	**82.42**	**85.77**	**75.29**	**70.64**	**78.12**	**61.49**	–	–	–

The models in the top and middle groups are trained on a single dataset, while those in the bottom group are jointly trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ (our split) datasets, where same images are shared among three datasets but with more expressions than the top and middle groups

Bold values denote the best results within the same group of methods

Table 2 Ablation study of jointly training on three datasets

Method	Image-level	Instance-level	RefCOCO			RefCOCO+			RefCOCOg∗		
			val	testA	testB	val	testA	testB	val	testA	testB
Baseline			75.17	79.53	68.72	63.42	70.55	53.38	60.62	64.57	53.81
w/o instance-level	✓		77.82	81.34	70.94	66.25	73.18	55.91	63.84	67.12	57.46
w/o image-level		Triplet Loss (2)	78.49	81.92	71.35	67.04	74.80	57.32	64.58	67.79	58.31
w/o image-level		Contrastive Loss (3)	79.33	83.28	72.64	67.81	75.77	58.63	65.73	68.54	58.89
Final w/$L_{ins-tri}$	✓	Triplet Loss (2)	80.61	84.19	73.71	69.28	76.72	59.80	68.26	69.81	61.34
Final w/L_{ins-cl}	✓	Contrastive Loss (3)	**82.42**	**85.77**	**75.29**	**70.64**	**78.12**	**61.49**	**69.92**	**71.75**	**62.68**
Final w/random sample	✓	Contrastive Loss (3)	80.93	84.30	73.65	69.47	76.61	59.87	68.48	70.04	61.52
Final w/same image sample	✓	Contrastive Loss (3)	82.05	85.52	74.89	70.38	77.82	61.17	69.46	71.32	62.23
Final w/o GCN	✓	Contrastive Loss (3)	81.40	84.26	74.52	69.72	77.18	60.57	69.13	70.71	61.67
Final w/E_v fixed	✓	Contrastive Loss (3)	81.56	84.41	74.19	69.30	76.83	60.17	68.64	70.02	60.92

The top and middle groups demonstrate the effectiveness of the proposed feature learning techniques in different levels, and the superiority of contrastive loss over triplet loss in the instance level. The bottom group shows the influence of the negative mining, GCN and co-training E_v in our model

Bold values denote the best results within the same group of methods

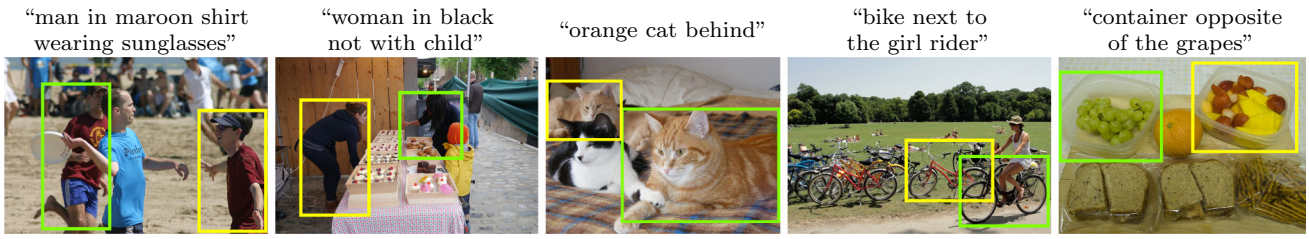

Fig. 3 Sample results of jointly training on the RefCOCO, RefCOCO+ and RefCOCOg* datasets. The green and yellow boxes represent the results of the baseline and our full model, respectively (Color figure online)

level loss improves the performance from the baseline model that does not consider the property of synonymous sentences. In the middle group of the table, the results are further improved in the full models with the loss on both levels, and the one with contrastive loss in the instance level achieves better performance than the one using triplet loss. In the bottom group, we provide the detailed ablation study in our model. We first show results of different negative sampling methods, including randomly sampling negatives from all images and sampling negatives from the same image. These results demonstrate that our negative mining performs better than naive random sampling by a larger margin. When removing the GCN from the model, the performance is slightly degraded compared to the full model in the middle group. This shows that our proposed feature learning techniques play the main role in performance improvement, while the proposed negative mining and GCN modules also help achieve better results. In our experiments, the entire model is trained end-to-end. To analyze the effect of co-training E_v, we conduct an experiment with E_v fixed during training. The results are shown in the last row of Table 2. We observe that the performance of training the entire model is better than training with E_v fixed. Furthermore, we present qualitative results in Fig. 3, which show that our full model better distinguishes between similar objects and understands relationships across objects.

4.4 Evaluation on the Unseen Dataset

To demonstrate that the proposed framework can transfer learned features to other unseen datasets, we use our trained models to evaluate on the Ref-Reasoning (Yang et al., 2020a) dataset, which contains completely different images and expressions from our training datasets. The results are shown in Table 3 and the performance of our model trained on the Ref-Reasoning dataset using the fully-supervised setting is provided as a reference. We first train our models on the RefCOCOg dataset (Nagaraja et al., 2016). When applying the intra-dataset loss in the full model, the performance is improved from the baseline and better than the two-stage MAttNet (Yu et al., 2018) and CM-Att-Erase (Liu et al., 2019) approaches, where we evaluate using their official pre-trained models. Then we jointly train our models on RefCOCO, RefCOCO+ and RefCOCOg∗ (our split) datasets. When more datasets are used to train our full model, the performance gains over the baseline increase, which demonstrates the effectiveness of our feature learning using synonymous sentences.

4.5 Transfer Learning on Unseen Datasets

To validate the feature learning ability of the proposed method, we conduct experiments on the transfer learning

Table 3 Evaluation on the Ref-Reasoning dataset with models trained on different datasets

Method	Training dataset	Number of objects		
		One	Two	Three
Ours (supervised)	Ref-reasoning	76.43	57.37	50.79
MAttNet (Yu et al., 2018)	RefCOCOg	49.81	32.17	25.83
CM-Att-Erase (Liu et al., 2019)	RefCOCOg	50.34	32.42	26.02
Ours (baseline)	RefCOCOg	50.26	31.41	26.16
Ours (full model)	RefCOCOg	**53.49**	**33.61**	**28.09**
Ours (baseline)	RefCOCO, RefCOCO+, RefCOCOg∗	54.56	33.11	28.46
Ours (full model)	RefCOCO, RefCOCO+, RefCOCOg∗	**57.96**	**36.82**	**32.61**

The models are trained on the Ref-Reasoning (top group), RefCOCOg (middle group), or RefCOCO, RefCOCO+ and RefCOCOg∗ datasets (bottom group)
Bold values denote the best results within the same group of methods

Table 4 Transfer learning on the Ref-Reasoning dataset with different settings of pre-training (Pre) and fine-tuning (FT)

Method	Pre-training Dataset	Our loss in		Number of objects		
		Pre	FT	One	Two	Three
SGMN (Yang et al., 2020a)∗				80.17	62.24	56.24
SGMN (Yang et al., 2020a)				73.86	54.03	45.69
Ours (baseline)				76.43	57.37	50.79
Ours (baseline)	RefCOCO, RefCOCO+, RefCOCOg*			76.54	57.43	50.81
Ours	RefCOCO, RefCOCO+, RefCOCOg*		✓	78.72	59.14	52.03
Ours	RefCOCO, RefCOCO+, RefCOCOg*	✓		79.16	59.47	52.39
Ours (full model)	RefCOCO, RefCOCO+, RefCOCOg*	✓	✓	**81.94**	**62.73**	**55.86**

The models in the top group are directly trained on Ref-Reasoning, while those in the bottom group are pre-trained on RefCOCO, RefCOCO+ and RefCOCOg∗, and fine-tuned on Ref-Reasoning. Note that ∗ in the first row indicates that SGMN (Yang et al., 2020a) uses ground truth proposals to generate final outputs, which is served as a reference here. The results of SGMN (second row) are generated from automatically detected objects
Bold values denote the best results within the same group of methods

setting, where the models are pre-trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets, and fine-tuned on the Ref-Reasoning (Yang et al., 2020a) or ReferItGame (Kazemzadeh et al., 2014) dataset. We also note that the SGMN (Yang et al., 2020a)∗ model in the first row of Table 4 uses ground truth proposals from the dataset, which is served as a reference, but is not a direct comparison with our method that predicts the locations without accessing ground truth bounding boxes. For fair comparisons, we evaluate SGMN with automatically detected objects using the same detector as ours. The detected objects are generated from Mask R-CNN with ResNet-101 pre-trained on COCO's training images, excluding those in the validation and testing sets of RefCOCO, RefCOCO+ and RefCOCOg. The results are presented in the second row of Table 4. With the same object detector and training data, our model has better performance than SGMN.

Table 4 shows that two models of "Ours (baseline)" with or without the pre-training stage, perform very similarly to each other on Ref-Reasoning (Yang et al., 2020a). These results show that it is challenging to learn transferable features by simply pre-training on existing datasets. However, by introducing our feature learning schemes, either during pre-training (Pre) or fine-tuning (FT), our models achieve better performance. When using our method in both pre-training and fine-tuning stages, the performance is further improved.

Similar comparisons can be observed in Table 5, where the models are fine-tuned on the ReferItGame (Kazemzadeh et al., 2014) dataset. In the bottom group, we demonstrate that the feature learning technique improves the performance

Table 5 Transfer learning on the ReferItGame dataset with different settings of pre-training (Pre) and fine-tuning (FT)

Method	Pre-training Dataset	Our loss in		Split
		Pre	FT	test
ZSGNet (Sadhu et al., 2019)				58.63
Darknet-LSTM (Yang et al., 2019b)				58.76
Darknet-BERT (Yang et al., 2019b)				59.30
RCCF (Liao et al., 2020)				63.79
ReSC-Large (Yang et al., 2020c)				64.60
LBYL-Net-LSTM (Huang et al., 2021)				65.48
LBYL-Net-BERT (Huang et al., 2021)				67.47
Ours (baseline)				57.04
Ours (baseline)	RefCOCO, RefCOCO+, RefCOCOg∗			57.13
Ours	RefCOCO, RefCOCO+, RefCOCOg∗		✓	58.54
Ours	RefCOCO, RefCOCO+, RefCOCOg∗	✓		58.89
Ours (full model)	RefCOCO, RefCOCO+, RefCOCOg∗	✓	✓	**61.71**

The models in the top group are directly trained on ReferItGame, while those in the bottom group are pre-trained on RefCOCO, RefCOCO+ and RefCOCOg∗, and fine-tuned on ReferItGame
Bold value denotes the best results within the same group of methods

consistently by using the proposed loss terms. Compared to results in the top group, despite that our baseline model does not perform better than existing methods, we show significant improvement by using our proposed loss functions to achieve better performance. This shows the ability of our method for transferring learned representations to other datasets.

In Table 1, we have showed that our method has better or competitive performance to RCCF (Liao et al., 2020), ReSC (Yang et al., 2020c) and LBYL-Net (Huang et al., 2021). One reason for their higher performance on ReferItGame is that they do not have the proposal generation step in the object detector, which is preferred for ReferItGame specifically. That is, ReSC and LBYL-Net use a one-stage object detector (Darknet-53), and RCCF directly generates the object size and offset by regression, which are different from our two-stage object detector (Mask R-CNN). Such a performance gap on ReferItGame can also be observed in previous methods that use two-stage object detectors and is also pointed out in the papers of RCCF, ReSC and LBYL-Net, in which handling such a difference is out of the scope of this paper. Despite the constraint in the object detector, we show in Table 5 that by applying the proposed contrastive loss, we can improve the performance from our baseline by 4.5% on ReferItGame, which is similar to the improvement on Ref-Reasoning (2–4% in Table 3 and 5% in Table 4).

4.6 Analysis of Learned Features

To demonstrate the effectiveness of the proposed loss, we calculate the similarity (dot product) between the instance-level features of synonymous sentences when jointly training on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets. We randomly sample a pair of synonymous sentences for each image, and compute the average value of all samples in the validation and testing splits of each dataset. Table 6 shows the similarity scores on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets, while Table 7 provides the similarity computed on the Ref-Reasoning and ReferItGame datasets in the transfer learning setting. From the results, we observe that the similarity in the embedding space of synonymous sentences is higher when the proposed loss terms are applied in all the settings, which shows that our model is able to transfer the learned features to unseen datasets.

Table 6 Similarity between instance-level features of synonymous sentences

Method	RefCOCO	RefCOCO+	RefCOCOg∗
Ours (baseline)	0.884	0.873	0.851
Ours (full model)	0.926	0.911	0.885

The models are jointly trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets

Table 7 Similarity between instance-level features of synonymous sentences

Method	Fine-tune	Ref-Reasoning	ReferItGame
Ours (baseline)		0.762	0.787
Ours (full model)		0.781	0.804
Ours (baseline)	✓	0.843	0.856
Ours (full model)	✓	0.881	0.892

All models are jointly trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets. Only models in the bottom group are fine-tuned on Ref-Reasoning or ReferItGame

4.7 Visualization of Response Maps

To demonstrate the effectiveness of the proposed feature learning technique, we show the response maps generated by our model in Fig. 4. The three expressions are anchor, positive and negative samples respectively. The anchor and positive sample focus on a similar region, while the negative sample attends to a different region.

4.8 Qualitative Results

In Figs. 5, 6, and 7, we present more qualitative results generated by our full model trained on the RefCOCO, RefCOCO+ and RefCOCOg∗ datasets. The proposed method is able to localize objects accurately given the synonymous sentences with paraphrases and also distinguish between sentences that describe different objects. We also provide some failure cases

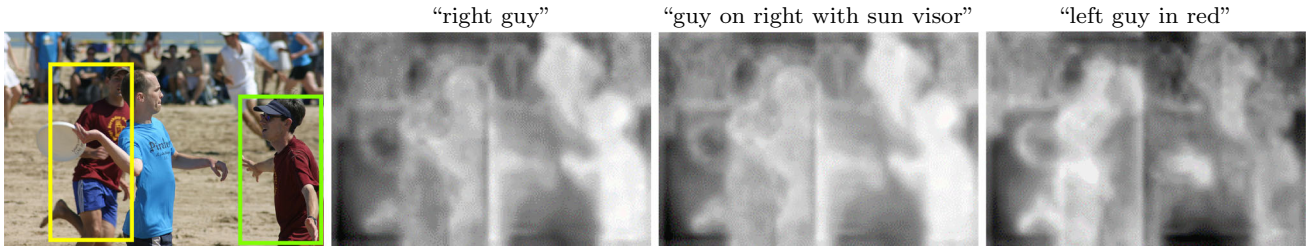

Fig. 4 Visualization of response maps. The green box represents the anchor and positive object, while the yellow box indicates the negative object. The three expressions are anchor, positive and negative samples respectively. The anchor and positive expression focus on a similar region, while the negative sample attends to another region (Color figure online)

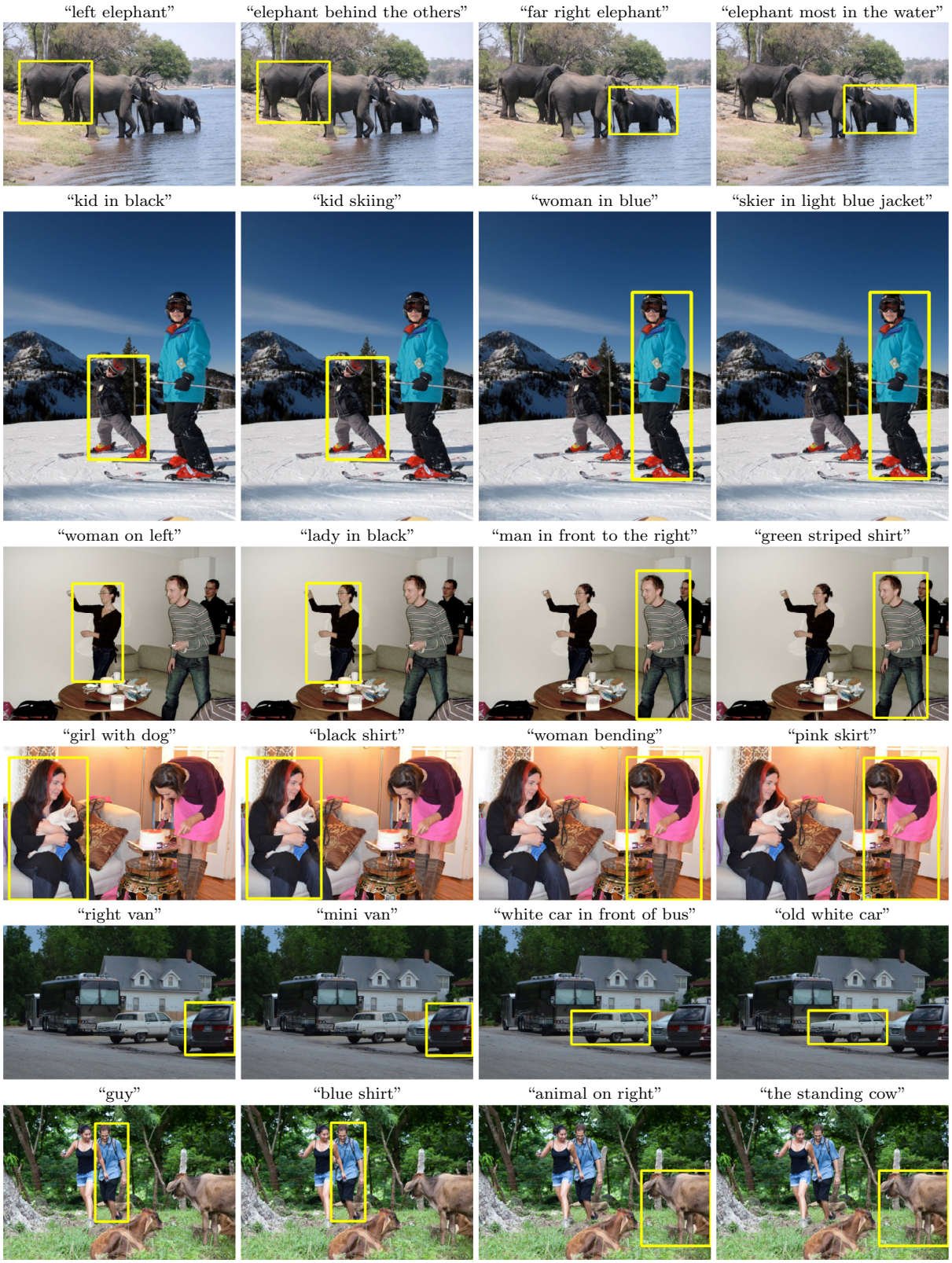

Fig. 5 Sample results of jointly training on the RefCOCO, RefCOCO+ and RefCOCOg* datasets. The yellow boxes are the results of our full model (Color figure online)

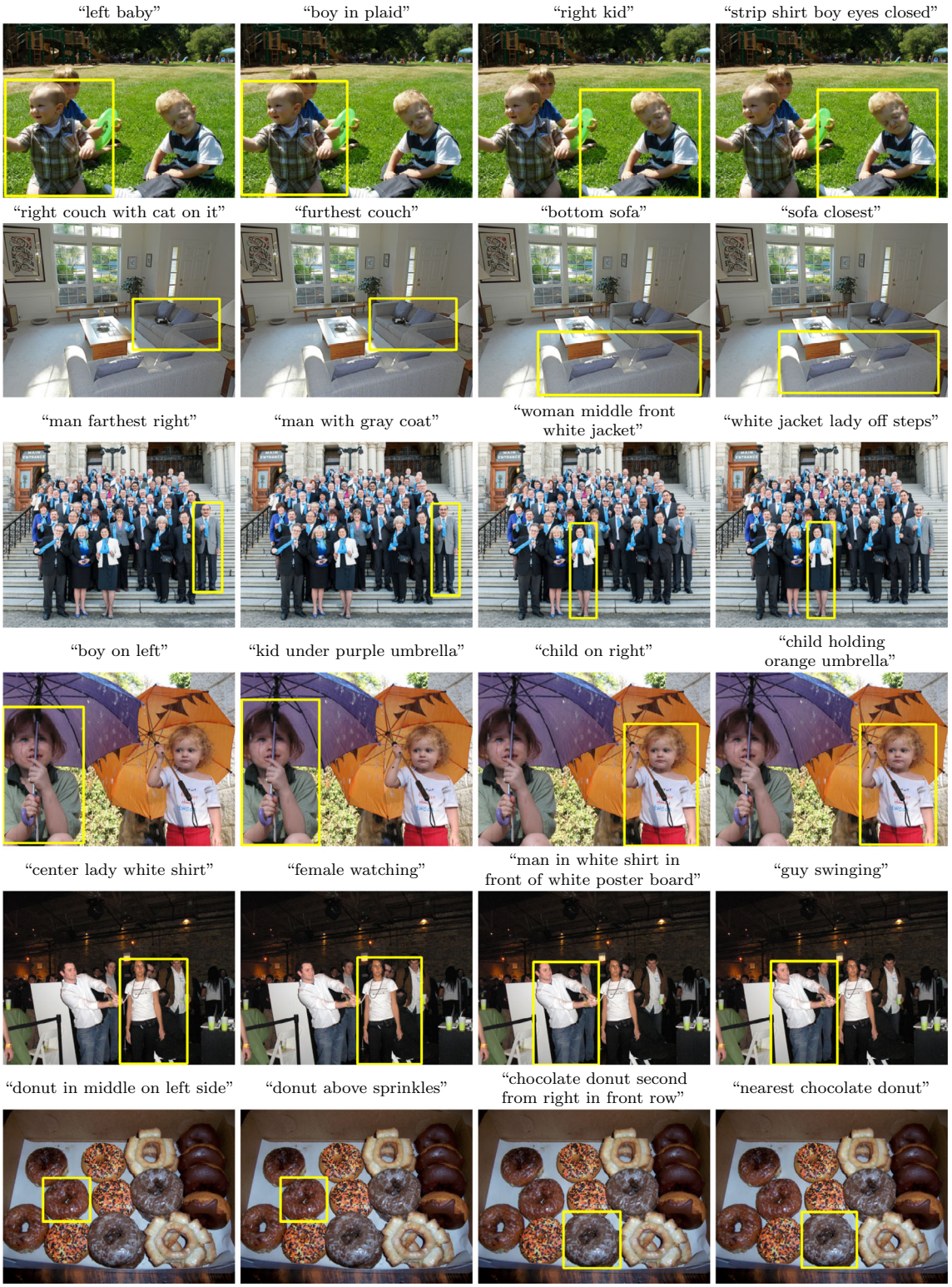

Fig. 6 Sample results of jointly training on the RefCOCO, RefCOCO+ and RefCOCOg* datasets. The yellow boxes are the results of our full model (Color figure online)

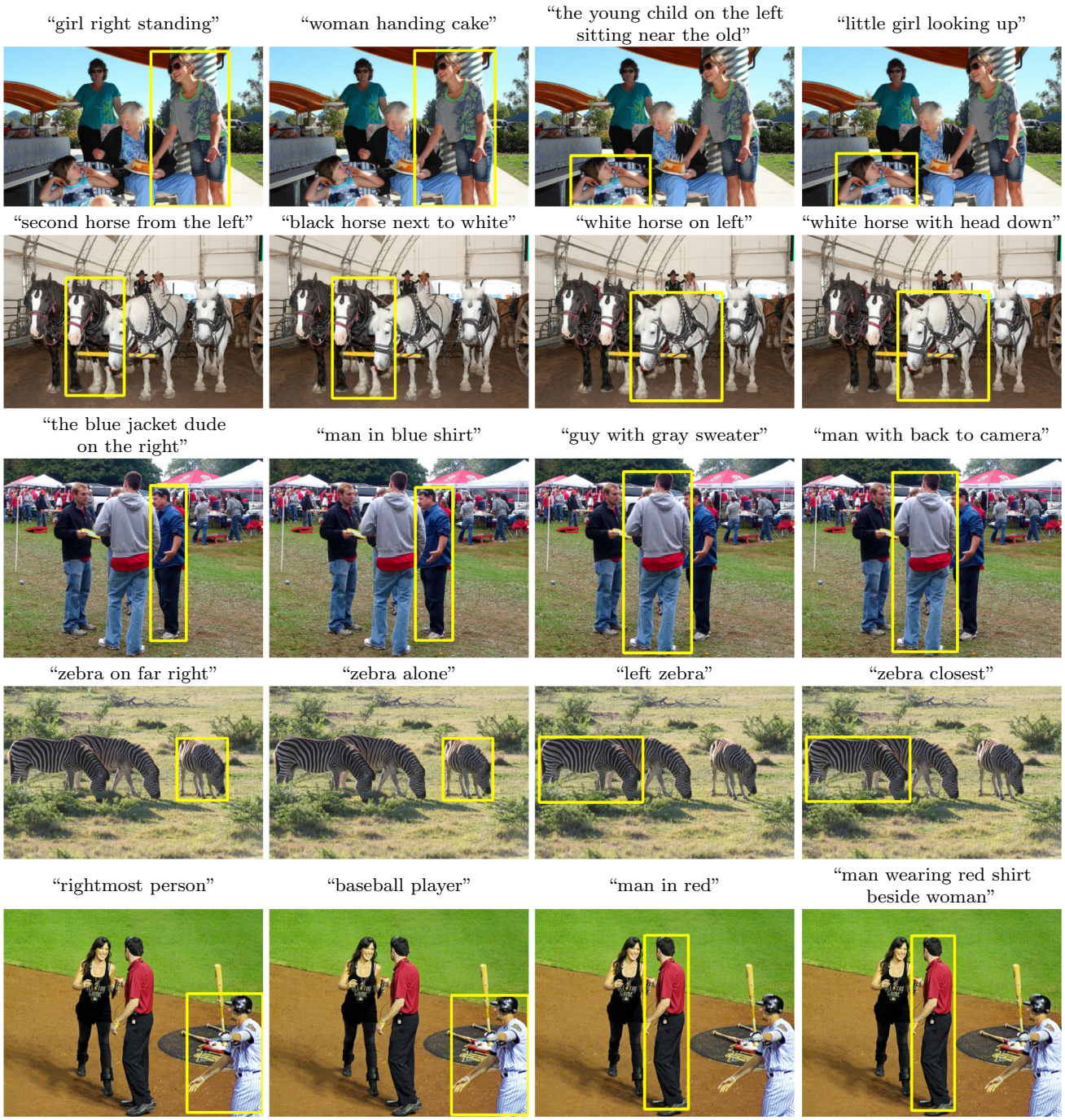

Fig. 7 Sample results of jointly training on the RefCOCO, RefCOCO+ and RefCOCOg* datasets. The yellow boxes are the results of our full model (Color figure online)

Fig. 8 Failure cases of our method. The red and yellow boxes represent the ground truth and our results, respectively (Color figure online)

of our method in Fig. 8. While the proposed algorithm shows the effectiveness on referring expression comprehension, it still suffers from some unfavorable effects, such as objects sharing similar attributes or ambiguous sentences.

4.9 Limitations and Discussion

Our model operates on the assumption that multiple expressions are available to describe one object. However, we note that most existing datasets have multiple sentences annotated for each object, due to the nature of how referring expressions are generated. For example, these expressions can be automatically generated by expression templates and scene graphs (Yang et al., 2020a). In addition, if there is a new dataset without multiple expressions, our model can still transfer the features learned from existing datasets to the new dataset. In Sects. 4.4 and 4.5, we show such benefit when evaluating on the unseen dataset and in the transfer learning setting.

5 Conclusions

In this paper, we focus on the task of referring expression comprehension and tackle the challenge caused by the varieties of synonymous sentences. To deal with this problem, we propose an end-to-end trainable framework that considers the property in languages for paraphrasing the objects to learn contrastive features. To this end, we employ the feature learning techniques on the image level as well as the instance level to encourage language features describing the same object to attend closely in the visual embedding space, while the expressions identifying different objects to be separated. We design two negative mining strategies to further facilitate the learning process. Extensive experiments and the ablation study on multiple referring expression datasets demonstrate the effectiveness of the proposed algorithm. Moreover, in the cross-dataset and transfer learning settings, we show that the proposed method is able to transfer learned representations to other datasets.

References

Chen, L., Ma, W., Xiao, J., Zhang, H., & Chang, SF. (2021). Ref-NMS: Breaking proposal bottlenecks in two-stage referring expression grounding. In *AAAI*.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020a). A simple framework for contrastive learning of visual representations. In *ICML*.

Chen, YC., Li, L., Yu, L., Kholy, AE., Ahmed, F., Gan, Z., Cheng, Y., & Liu, J. (2020b). Uniter: Universal image-text representation learning. In *ECCV*.

Chen, Y.W., Tsai, Y.H., Wang, T., Lin, Y.Y., & Yang, M.H. (2019). Referring expression object segmentation with caption-aware consistency. In *BMVC*.

Chen, Z., Wang, P., Ma, L., Wong, K.Y.K., & Wu, Q. (2020c). Cops-Ref: A new dataset and task on compositional referring expression comprehension. In *CVPR*.

Deng, J., Yang, Z., Chen, T., Zhou, W., & Li, H. (2021). TransVG: End-to-end visual grounding with transformers. In *ICCV*.

Devlin, J., Chang, MW., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805

Escalante, H. J., Hernndez, C. A., Gonzalez, J. A., López-López, A., Montes, M., Morales, E. F., Sucar, L. E., Villaseñor, L., & Grubinger, M. (2010). The segmented and annotated IAPR TC-12 benchmark. *CVIU, 114*(4), 419–428.

Gan, Z., Chen, YC., Li, L., Zhu, C., Cheng, Y., & Liu, J. (2020). Large-scale adversarial training for vision-and-language representation learning. In *NeurIPS*.

Hadsell, R., Chopra, S., & LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *CVPR*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

He, K., Gkioxari, G., Dollar, P., & Girshick, R. (2017). Mask R-CNN. In *ICCV*.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Hu, R., Xu, H., Rohrbach, M., Feng, J., Saenko, K., & Darrell, T. (2016). Natural language object retrieval. In *CVPR*.

Huang, B., Lian, D., Luo, W., & Gao, S. (2021). Look before you leap: Learning landmark features for one-stage visual grounding. In *CVPR*.

Hudson, D.A., & Manning, C.D. (2019). Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*.

Kalantidis, Y., Sariyildiz, MB., Pion, N., Weinzaepfel, P., & Larlus, D. (2020). Hard negative mixing for contrastive learning. In *NeurIPS*

Kazemzadeh, S., Ordonez, V., Matten, M., & Berg, T.L. (2014). ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised contrastive learning. In *NeurIPS*.

Kipf, TN., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, X., Yin, X., Li, C., Hu, X., Zhang, P., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., Choi, Y., & Gao, J. (2020). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*.

Liao, Y., Liu, S., Li, G., Wang, F., Chen, Y., Qian, C., & Li, B. (2020). A real-time cross-modality correlation filtering method for referring expression comprehension. In *CVPR*.

Lin, TY., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, CL. (2014). Microsoft COCO: Common objects in context. In *ECCV*.

Liu, J., Wang, L., & Yang, M.H. (2017). Referring expression generation and comprehension via attributes. In *ICCV*.
Liu, X., Wang, Z., Shao, J., Wang, X., & Li, H. (2019). Improving referring expression grounding with cross-modal attention-guided erasing. In *CVPR*.
Lu, J., Batra, D., Parikh, D., & Lee, S. (2019). ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.
Lu, J., Goswami, V., Rohrbach, M., Parikh, D., & Lee, S. (2020). 12-in-1: Multi-task vision and language representation learning. In *CVPR*.
Luo, R., & Shakhnarovich, G. (2017). Comprehension-guided referring expressions. In *CVPR*.
Mao, J., Huang, J., Toshev, A., Camburu, O., Yuille, A.L., & Murphy, K. (2016). Generation and comprehension of unambiguous object descriptions. In *CVPR*.
Nagaraja, VK., Morariu, V.I., & Davis, L.S. (2016). Modeling context between objects for referring expression understanding. In *ECCV*.
Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In *NAACL*.
Rohrbach, A., Rohrbach, M., Hu, R., Darrell, T., & Schiele, B. (2016). Grounding of textual phrases in images by reconstruction. In *ECCV*.
Sadhu, A., Chen, K., & Nevatia, R. (2019). Zero-shot grounding of objects from natural language queries. In *ICCV*.
Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *CVPR*.
Sohn, K. (2016). Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*.
van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv:1807.03748
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Lu., & Polosukhin, I. (2017). Attention is all you need. In *NeurIPS*.
Wang, L., Li, Y., & Lazebnik, S. (2016). Learning deep structure-preserving image-text embeddings. In *CVPR*.
Yamaguchi, M., Saito, K., Ushiku, Y., & Harada, T. (2017). Spatio-temporal person retrieval via natural language queries. In *ICCV*.
Yang, S., Li, G., & Yu, Y. (2019a). Dynamic graph attention for referring expression comprehension. In *ICCV*.
Yang, S., Li, G., & Yu, Y. (2020a). Graph-structured referring expressions reasoning in the wild. In *CVPR*.
Yang, S., Li, G., &Yu, Y. (2020b). Propagating over phrase relations for one-stage visual grounding. In *ECCV*.
Yang, Z., Gong, B., Wang, L., Huang, W., Yu, D., & Luo, J. (2019b). A fast and accurate one-stage approach to visual grounding. In *ICCV*.
Yang, Z., Chen, T., Wang, L., & Luo, J. (2020c). Improving one-stage visual grounding by recursive sub-query construction. In *ECCV*.
Yu, L., Poirson, P., Yang, S., Berg, A.C., & Berg, T.L. (2016). Modeling context in referring expressions. In *ECCV*.
Yu, L., Tan, H., Bansal, M., & Berg, T.L. (2017). A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*.
Yu, L., Lin, Z., Shen, X., Yang, J., Lu, X., Bansal, M., & Berg, T.L. (2018). MAttNet: Modular attention network for referring expression comprehension. In *CVPR*.
Zhang, H., Niu, Y., & Chang, S.F. (2018). Grounding referring expressions in images by variational context. In *CVPR*.
Zhou, L., Palangi, H., Zhang, L., Hu, H., Corso, J.J., & Gao, J. (2020). Unified vision-language pre-training for image captioning and vqa. In *AAAI*.
Zhuang, B., Wu, Q., Shen, C., Reid, I., & Hengel, A. (2018). Parallel attention: A unified framework for visual object discovery through dialogs and queries. In *CVPR*.