Appendix A. Results on RefCOCO

The extended experiments on RefCOCO [48] are reported in Table 4. RefCOCO contains 19,994 images and 50,000 referred objects originally from MSCOCO [19], with 142,210 collected referring expressions. The referred objects are selected from MSCOCO annotations, and are in one of the 80 object classes defined by MSCOCO with no free-form expressions. We follow the split [48] of train/validation/testA/testB, which has 120,624, 10,834, 5,657 and 5,095 expressions, respectively. "testA" contains images with multiple people and "testB" contains images with instances of all other objects.

Organizing the results in the same way as Tables 1 and 2, the state-of-the-art results [49, 50, 47] are reported in the top portion of Table 4. The middle contains the variants of the similarity network [42], and the bottom shows our results. Similarly, we list the region proposals used by two-stage methods in the "Region Proposals" column. For studies [49, 50, 47] that use different proposals during training and inference, we show the "Region Proposals" in a format of "A/B" where A stands for the proposals used during training and B during inference. COCO-trained ResNet-101 is used as "visual features" and LSTM is used for "language embedding" in all reported results unless stated otherwise in method names.

Ours-LSTM outperforms the state-of-the-art methods except MAttNet [47], which uses extra supervision such as attributes and class labels of region proposals. Compared to the improvements from two-stage methods [42] on ReferitGame (25.0%) and Flickr30K Entities (7.8%), the improvement on RefCOCO is rather small (1.4%). We prove empirically that the major reason is the good proposal quality on RefCOCO. First, the hit rate analyses in Table 5 show that the proposals generated by the COCO-trained proposal networks are sufficiently good on RefCOCO. The used proposals cover almost all the referred objects on RefCOCO (92.4% for detection and 98.5% for region proposals). This is because the images and objects in RefCOCO are a subset of COCO. Second, among the two-stage methods in the middle of Table 5, the similarity network with COCO-trained Faster R-CNN [34] generated proposals significantly outperforms (by 14.1%) the one using Edgebox [54]. This shows the strong correlation between the good proposal quality and the good performance of two-stage methods on RefCOCO.

Since RefCOCO is a subset of COCO and has shared images and objects, the proposal quality with COCO-trained proposal networks nearly perfect as shown in Table 5. With such ideal proposals on RefCOCO, two-stage methods can greatly narrow the performance gap between one- and two-stage methods. However, this special case only holds on the subset of COCO. Both the hit rate and grounding accuracy drop dramatically when such proposal networks are directly used on other datasets [15, 30]. Table 3 reports a lower hit rate of region proposal networks (RPNs) generated proposals compared to Edgebox, which is contradictory to the analyses on RefCOCO in Table 5. Similarly, two-stage methods with RPNs generated proposals perform worse than those with Edgebox. On ReferItGame, similarity network with RPNs generated proposals generates an accuracy of 27.1%, compared to the 34.5% with Edgebox. This posts a caveat of the RefCOCO datasets that free-form expressions might be necessary. We hope future works will experiment both with and beyond COCO.

Regarding the problem of imperfect region candidates in two-stage methods, a natural idea is end-to-end fine-tuning region proposal networks (RPNs), which does boost the two-stage methods' overall performances. QRC Net [3] trains RPNs in an end-to-end manner and achieves the following results on ReferitGame (Sim. Net [42]: 34.54%, QRC Net [3]: 44.07%, Ours: 59.30%) and Flickr30K Entities (Sim. Net [42]: 60.89%, QRC Net [3]: 65.14%, Ours: 68.69%). Besides, the two-stage methods perform better on RefCOCO (cf. Tables 4 and 5) than them on the other two datasets because their RPNs are trained not only by the mentions in the referring expressions but also other COCO objects. Nonetheless, our approach still gives rise to better overall results (as well as the faster inference speed and simpler framework).

Table 4. Referring expression comprehension results on RefCOCO [48]. LSTM and COCO-trained Res101 are the encoders unless stated otherwise in method names.

Method	Region Proposals	val	testA	testB
SLR [49]	GT/FRCN Detc.	69.48	73.71	64.96
VC-VGG16 [50]	GT/SSD Detc. [21]	-	73.33	67.44
MAttNet Base [47]	GT/FRCN Detc.	72.72	76.17	68.18
MAttNet [47]	GT/FRCN Detc.	**76.40**	**80.43**	69.28
Similarity Net [42]	Edgebox N=200	57.33	57.22	55.60
Similarity Net [42]	GT/FRCN Detc.	71.48	74.90	67.32
Sim. Net-Darknet [42]	GT/FRCN Detc.	72.27	75.12	67.91
Ours-Darknet-Bert	None	72.05	74.81	67.59
Ours-Darknet-LSTM	None	73.66	75.78	**71.32**

Table 5. Hit rates of region proposals on RefCOCO.

Hit rate, N=200	val	testA	testB
FRCN Detc. [34]	92.42	95.83	88.87
FRCN RP [34]	98.52	**99.47**	97.60
Edgebox [54]	89.01	89.62	89.28
SS [41]	84.28	81.72	89.54
Ours	**98.80**	99.08	**98.64**

Appendix B. Cross-Sample Relationships

Inspired by previous studies [3, 5] that successfully exploit all phrases and queries on the same image for visual grounding, we extend our vanilla framework to utilize cross-sample relationships. Given an anchor sample with image I_i and query Q_i describing an object of interest, we

Table 6. Visual grounding results of cross-sample methods.

Method	ReferIt	Flickr30K	RefCOCO		
	Game	Entities	val	testA	testB
SeqGROUND [5]	-	61.60	-	-	-
QRC Net [3]	44.07	65.14	-	-	-
Ours	59.30	68.69	73.66	75.78	71.32
Ours-Cross sample	60.37	69.15	74.52	76.51	71.88

define samples in the positive bag as all other pairs with different queries describing the same object in the same image. For example in Figure 2, "two people sitting" and "two people in the middle of the boat" refer to the same region. The negative bag consists of the intra-image negative samples with the queries describing different regions in the same image, and the inter-image negative samples with completely different images. Given an anchor sample, we assume that the fused visual-textual feature should be more similar to the ones in the positive bag compared to the negative ones. The proposed feature regularization enforces such relationship with a triplet loss:

$$L_{reg} = \sum_i \big[\|f(I_i, Q_i) - f(I_{P_i}, Q_{P_i})\|_2^2 - \|f(I_i, Q_i) - f(I_{N_i}, Q_{N_i})\|_2^2 + m\big]_+$$

where (I_{P_i}, Q_{P_i}) and (I_{N_i}, Q_{N_i}) are the sampled positive and negative image-query pairs. The feature f can be any fused visual textual feature. In this study, we define f as the average pooling results of the feature in the last but one layer.

In experiments, we set margin $m = 1$ and regularization term weight $w_{reg} = 1$. Table 6 reports the performance with feature regularization. We observe an improvement in performance on all three datasets [15, 30, 48]. We also experiment with hard triplet generation, but observe no major change in performances.