Supplementary Material: Referring Expression Object Segmentation with Caption-Aware Consistency

Yi-Wen Chen[1]
chenyiwena@gmail.com
Yi-Hsuan Tsai[2]
ytsai@nec-labs.com
Tiantian Wang[3]
tiantianwang.ice@gmail.com
Yen-Yu Lin[1]
yylin@citi.sinica.edu.tw
Ming-Hsuan Yang[3][4]
mhyang@ucmerced.edu

[1] Academia Sinica
[2] NEC Laboratories America
[3] University of California, Merced
[4] Google Cloud

In this supplementary document, we provide additional experimental results, including 1) sensitivity analysis of the hyperparameter in the objective function, 2) training/inference time performance, and 3) more quantitative and qualitative results of our method for referring expression object localization and segmentation.

1 Hyperparameter Analysis

In the objective function (7) of the main paper, the hyperparameter α controls the importance of the caption-aware consistency loss, which links referring expression comprehension and generation. Table 1 reports the localization performance of our method by setting α to 0.01, 0.1, and 1, respectively. It can be observed that our method is robust to the value of α. Based on the results in Table 1, we set α to 0.1 in all our experiments.

Table 1: Localization results with different weights of the caption-aware consistency loss.

	α	0.01	0.1	1
RefCOCO	val	77.04	**77.08**	76.74
	testA	79.90	**80.34**	80.10
	testB	70.28	**70.62**	70.48
RefCOCOg	val*	62.02	**62.34**	62.11
	val	65.56	65.83	**65.87**
	test	65.27	**65.44**	65.32

2 Training Time and Runtime Analysis

Our method is implemented using PyTorch on a machine with an Intel Xeon 2.5 GHz processor and an NVIDIA GTX 1080 Ti GPU with 11 GB memory. The proposed framework is composed of two networks, *i.e.*, the comprehension and generation networks. We report the training time in individual networks and the joint framework with a batch size as 1 in Table 2. During inference, the proposed method takes 0.171 seconds to process one image in a single forward pass, while the state-of-the-art method MAttNet [12] requires multiple steps and takes 0.671 seconds for an image, which is almost 4 times of ours, shown in Table 3.

Table 2: Training time per iteration of the proposed method.

Network	Time (s)
Comprehension network	0.334
Generation network	0.244
Joint framework	0.402

Table 3: Runtime performance.

Method	Time (s)
MAttNet [12]	0.671
Ours	0.171

3 Localization and Segmentation Results

We provide results on the RefCOCO+ [10] dataset in Table 4, with results on the other two datasets originally reported in the main paper. We notice that the relative improvement made

Table 4: Localization results of our method and the competing methods on three datasets. We summarize the major information used in each method, including context (C), attribute prediction (Attr), attention module (Attn), location (L), relations between objects (R), and joint training with referring expression generation (J).

		RefCOCO			RefCOCO+			RefCOCOg		
Model	Info.	val	testA	testB	val	testA	testB	val*	val	test
Nagaraja *et al.* [8]	C	57.30	58.60	56.40	-	-	-	-	-	49.50
Luo *et al.* [6]	J	-	67.94	55.18	-	57.05	43.33	49.07	-	-
Liu *et al.* [5]	Attr, J	-	72.08	57.29	-	57.97	46.20	52.35	-	-
Yu *et al.* [11]	J	-	73.78	63.83	-	60.48	49.36	59.84	-	-
MAttNet [12]	Attr, Attn, L, R	76.65	81.14	69.99	65.33	71.62	56.02	-	66.58	67.27
VC [13]	C	-	73.33	67.44	-	58.40	53.18	62.30	-	-
baseline	-	72.65	76.65	65.75	59.22	66.26	47.68	54.18	58.09	58.32
+ spatial coords [2]	L	75.89	78.57	68.54	61.12	68.54	49.01	61.37	64.10	64.21
+ spatial filters	L	76.98	79.30	69.75	61.74	**69.46**	50.91	61.65	65.18	65.28
+ caption consistency	J	76.05	78.84	69.36	61.29	68.46	49.85	60.69	64.71	63.79
full model	L, J	**77.08**	**80.34**	**70.62**	**62.15**	69.26	**51.32**	**62.34**	**65.83**	**65.44**

Table 5: Localization results of our method and the competing methods on three datasets. All the methods use the VGG network as the feature extractor. We summarize the major information used in each method, including context (C), attribute prediction (Attr), location (L), and joint training with referring expression generation (J).

			RefCOCO			RefCOCO+			RefCOCOg		
Model	Network	Info.	val	testA	testB	val	testA	testB	val*	val	test
Nagaraja *et al.* [8]	VGG-16	C	57.30	58.60	56.40	-	-	-	-	-	49.50
Luo *et al.* [6]	VGG-16	J	-	67.94	55.18	-	57.05	43.33	49.07	-	-
Liu *et al.* [5]	VGG-19	Attr	-	72.08	57.29	-	57.97	46.20	52.35	-	-
Yu *et al.* [11]	VGG-16	J	-	73.78	63.83	-	60.48	49.36	59.84	-	-
VC [13]	VGG-16	C	-	73.33	67.44	-	58.40	**53.18**	**62.30**	-	-
Ours (spatial filters)	VGG-16	L	**72.01**	**75.27**	**68.30**	55.04	**61.41**	44.87	55.41	58.23	**58.52**

Table 6: Segmentation performance on RefCOCO.

Split	Model	Backbone	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9	IoU
val	D+RMI+DCRF [4]	Deeplab101	42.99	33.24	22.75	12.11	2.23	45.18
	MAttNet [12]	Res101	75.16	72.55	67.83	54.79	16.81	56.51
	RRN+LSTM [3]	Deeplab101	60.19	50.19	38.32	23.87	5.66	54.26
	RRN+LSTM+DCRF [3]	Deeplab101	61.66	52.50	42.40	28.13	8.51	55.33
	DMN [7]	DPN92	-	-	-	-	-	49.78
	Ours	Res101	75.97	72.96	67.98	54.48	17.47	58.90
testA	D+RMI+DCRF [4]	Deeplab101	42.99	33.59	23.69	12.94	2.44	45.69
	MAttNet [12]	Res101	79.55	77.60	72.53	59.01	13.79	62.37
	RRN+LSTM+DCRF [3]	Deeplab101	64.13	54.66	44.37	29.15	8.08	57.26
	DMN [7]	DPN92	65.83	57.82	46.80	27.64	5.12	54.83
	Ours	Res101	78.96	76.37	71.95	57.66	13.63	61.77
testB	D+RMI+DCRF [4]	Deeplab101	44.99	32.21	22.69	11.84	2.65	45.57
	MAttNet [12]	Res101	68.87	65.06	60.02	48.91	21.37	51.70
	RRN+LSTM+DCRF [3]	Deeplab101	59.35	50.32	39.82	27.30	10.05	53.95
	DMN [7]	DPN92	-	-	-	-	-	45.13
	Ours	Res101	68.95	65.63	60.65	49.72	20.18	53.81

by our method on RefCOCO+ is smaller than the ones on other two datasets. The main reason is that the location information of referring expressions is forbidden on this dataset, in which our spatial-aware dynamic filters may not be fully utilized to propagate useful knowledge from the language encoder. In addition, we report the performance of the proposed method using the Faster R-CNN framework with the VGG-16 network, and compare with other VGG-based models in Table 5. The results show that our VGG-based model with only the location information via spatial-aware dynamic filters performs favorably against other methods.

In Table 6, 7, and 8, we evaluate the segmentation quality by computing the IoU between predicted segments and ground truths. We consider the prediction with the IoU over 0.5 as correct ones. By setting five different thresholds from 0.5 to 0.9, we generate five results, which are represented as Pr@0.5, Pr@0.6, Pr@0.7, Pr@0.8, Pr@0.9, respectively. Overall, our method consistently and significantly outperforms other segmentation-based approaches that use a similar backbone network (*i.e.*, Deeplab [1] with ResNet-101) as ours. Similar to the localization results, MAttNet [12] that fuses multiple cues performs competitively with our model. In addition, we find that our model can still maintain a higher precision when the IoU criteria become more strict, *e.g.*, larger than 0.7. The reason is that our method is built upon a proposal-based network, which first tackles a simpler task of object localization and then performs segmentation.

We provide more results generated by our method and its two variants on the RefCOCO,

Table 7: Segmentation performance on RefCOCO+.

Split	Model	Backbone	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9	IoU
val	D+RMI+DCRF [4]	Deeplab101	20.52	14.02	8.46	3.77	0.62	29.86
	MAttNet [12]	Res101	64.11	61.87	58.06	47.42	14.16	46.67
	RRN+LSTM+DCRF [3]	Deeplab101	37.32	28.96	20.31	11.33	2.66	39.75
	DMN [7]	DPN92	-	-	-	-	-	38.88
	Ours	Res101	60.79	58.51	54.84	44.38	13.62	44.56
testA	D+RMI+DCRF [4]	Deeplab101	21.22	14.43	8.99	3.91	0.49	30.48
	MAttNet [12]	Res101	70.12	68.48	63.97	52.13	12.28	52.39
	RRN+LSTM+DCRF [3]	Deeplab101	40.80	31.66	22.74	12.78	2.78	42.15
	DMN [7]	DPN92	-	-	-	-	-	44.22
	Ours	Res101	68.23	66.43	62.43	49.95	12.09	50.03
testB	D+RMI+DCRF [4]	Deeplab101	20.78	14.56	8.80	4.58	0.80	29.50
	MAttNet [12]	Res101	54.82	51.73	47.27	38.58	17.00	40.08
	RRN+LSTM+DCRF [3]	Deeplab101	32.42	24.69	17.10	9.92	2.78	36.11
	DMN [7]	DPN92	-	-	-	-	-	32.29
	Ours	Res101	49.15	46.55	43.14	35.16	14.26	36.01

Table 8: Segmentation performance on RefCOCOg with two different splits.

Split	Model	Backbone	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9	IoU
val*	RRN+LSTM+DCRF [3]	Deeplab101	36.00	29.77	22.78	14.06	3.74	36.45
	KWAN [9]	Deeplab101	27.85	21.01	13.42	6.60	1.97	36.92
	DMN [7]	DPN92	-	-	-	-	-	36.76
	Ours	Res101	59.99	57.16	52.48	40.99	12.42	44.32
val	MAttNet [12]	Res101	64.48	61.52	56.50	43.97	14.67	47.64
	Ours	Res101	62.38	59.19	53.33	41.42	13.68	46.37
test	MAttNet [12]	Res101	65.60	62.92	56.50	43.97	14.67	48.61
	Ours	Res101	62.88	59.48	54.45	41.94	12.49	46.95

RefCOCO+, and RefCOCOg datasets. These results on the three datasets are shown in Figure 1, 2, and 3, respectively. We observe that the two variants, **baseline** and **spatial-aware filters**, sometimes fail to localize objects or cannot segment the entire object completely. In contrast, our method (**full model**) that considers the consistency between the query and output sentences, is able to localize objects accurately and generate more complete object segments. More results are presented in Figure 4, 5 and 6.

We also provide the failure cases of our method on the RefCOCO dataset in Figure 7. Though the proposed method shows its advantages over the state-of-the-art methods, it still suffers from some unfavorable effects when localizing and segmenting objects in challenging scenarios, such as objects sharing similar appearance (the first row of Figure 7), objects and their background having similar color distributions (the left images in the third row of Figure 7), and mutual occlusions among objects (the second row of Figure 7).

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv:1606.00915*, 2016.

[2] Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *ECCV*, 2016.

[3] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *CVPR*, 2018.

[4] Chenxi Liu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, and Alan Yuille. Recurrent multimodal interaction for referring image segmentation. In *ICCV*, 2017.

[5] Jingyu Liu, Liang Wang, and Ming-Hsuan Yang. Referring expression generation and comprehension via attributes. In *ICCV*, 2017.

[6] Ruotian Luo and Gregory Shakhnarovich. Comprehension-guided referring expressions. In *CVPR*, 2017.

[7] Edgar Margffoy-Tuay, Juan C. Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *ECCV*, 2018.

[8] Varun K. Nagaraja, Vlad I. Morariu, and Larry S. Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016.

[9] Hengcan Shi, Hongliang Li, Fanman Meng, and Qingbo Wu. Key-word-aware network for referring expression image segmentation. In *ECCV*, 2018.

[10] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C. Berg, and Tamara L. Berg. Modeling context in referring expressions. In *ECCV*, 2016.

[11] Licheng Yu, Hao Tan, Mohit Bansal, and Tamara L. Berg. A joint speaker-listener-reinforcer model for referring expressions. In *CVPR*, 2017.

[12] Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L. Berg. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*, 2018.

[13] Hanwang Zhang, Yulei Niu, and Shih-Fu Chang. Grounding referring expressions in images by variational context. In *CVPR*, 2018.

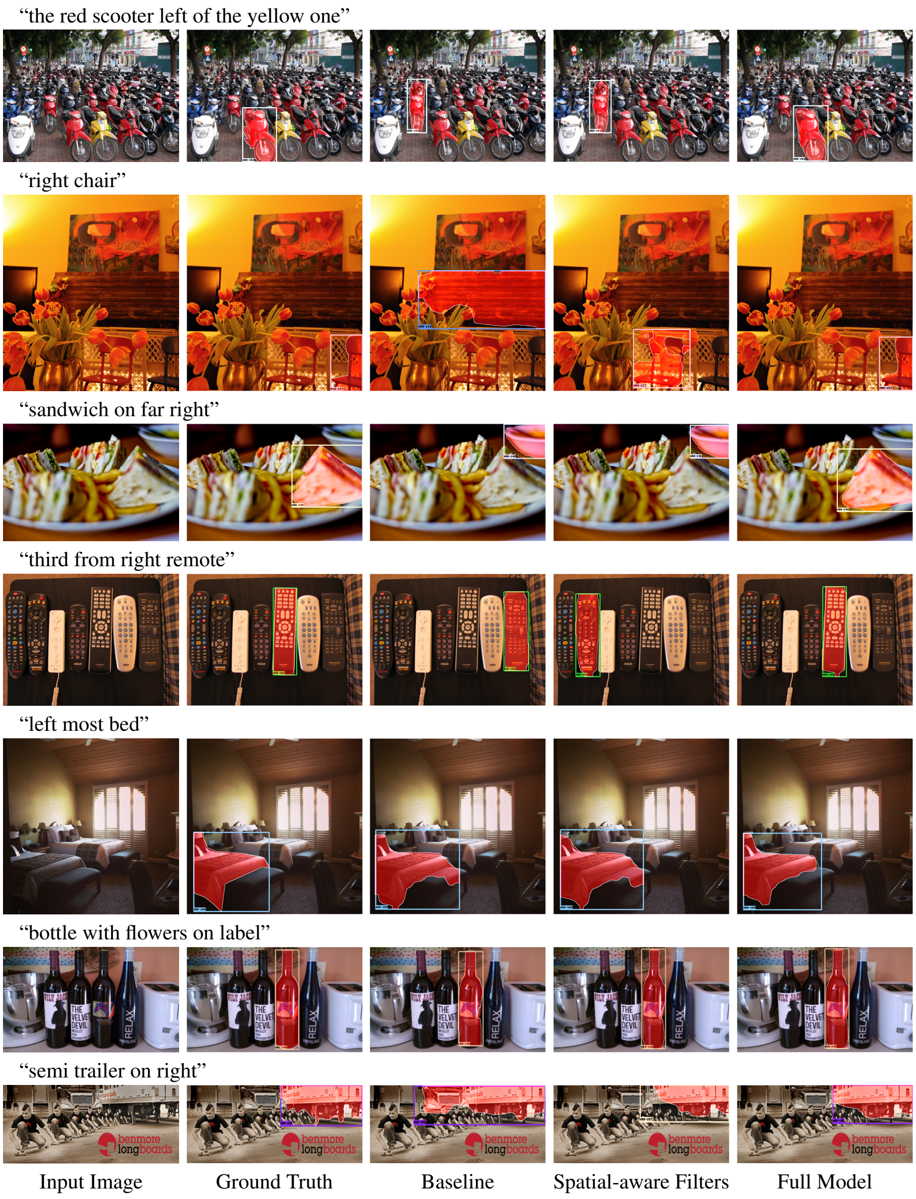

Figure 1: Sample results from different variants of the proposed model on the RefCOCO dataset.

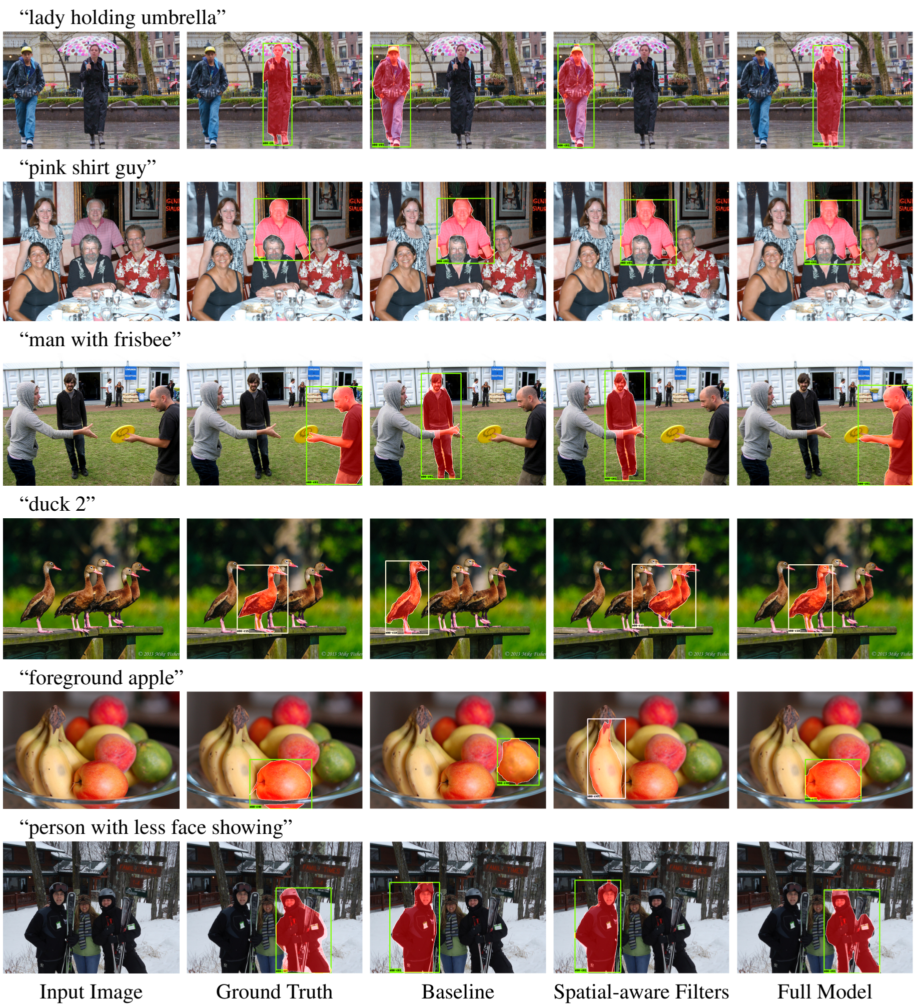

Figure 2: Sample results from different variants of the proposed model on the RefCOCO+ dataset.

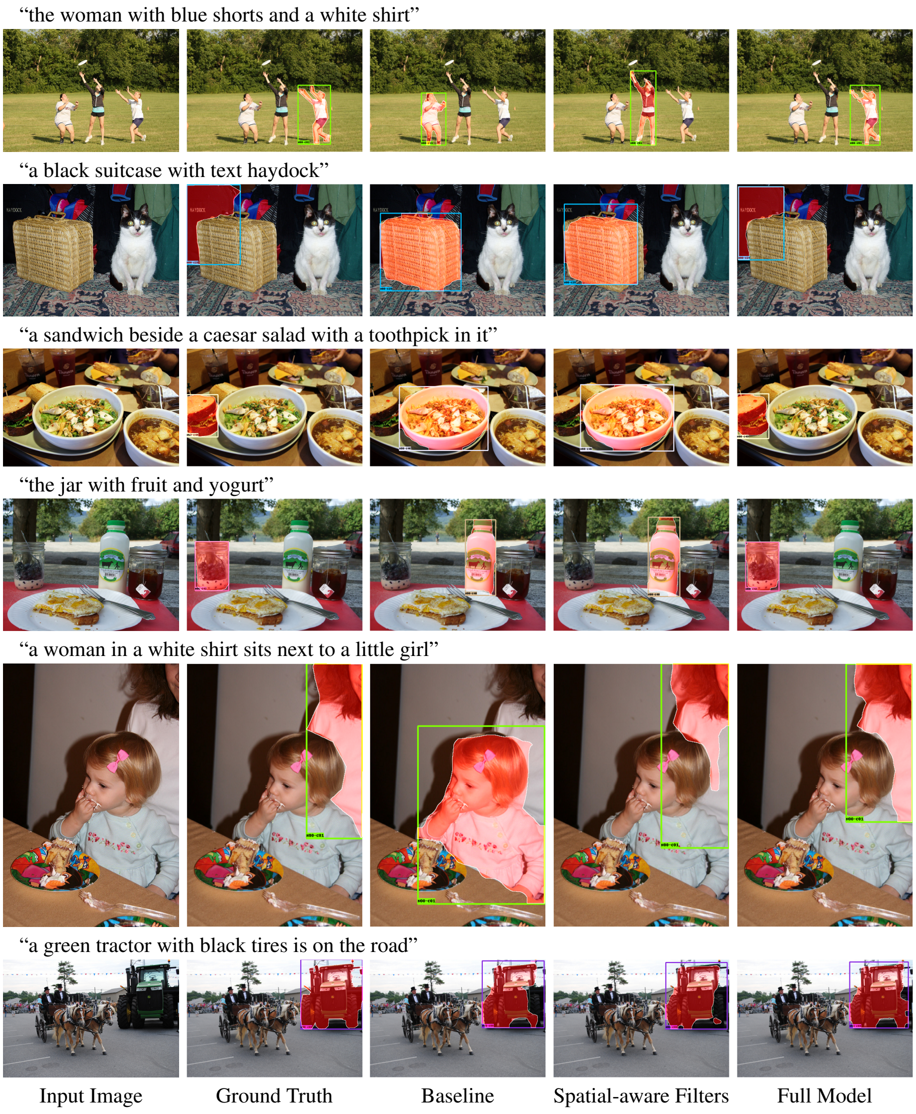
"the woman with blue shorts and a white shirt"
"a black suitcase with text haydock"
"a sandwich beside a caesar salad with a toothpick in it"
"the jar with fruit and yogurt"
"a woman in a white shirt sits next to a little girl"
"a green tractor with black tires is on the road"
Input Image
Ground Truth
Baseline
Spatial-aware Filters
Full Model

Figure 3: Sample results from different variants of the proposed model on the RefCOCOg dataset.

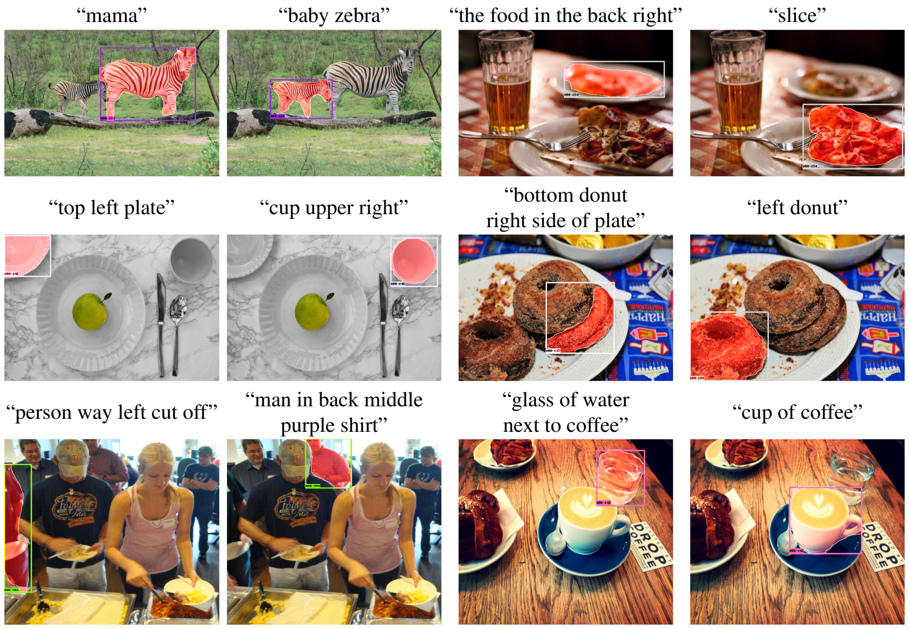
"mama"
"baby zebra"
"the food in the back right"
"slice"
"top left plate"
"cup upper right"
"bottom donut right side of plate"
"left donut"
"person way left cut off"
"man in back middle purple shirt"
"glass of water next to coffee"
"cup of coffee"

Figure 4: Sample results of objects referred by different query expressions on the RefCOCO dataset.

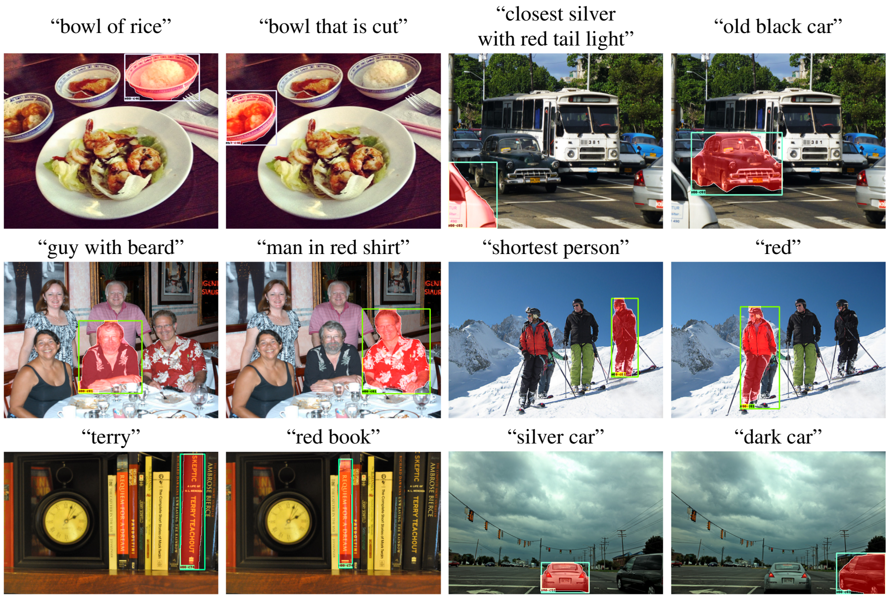

Figure 5: Sample results of objects referred by different query expressions on the RefCOCO+ dataset.

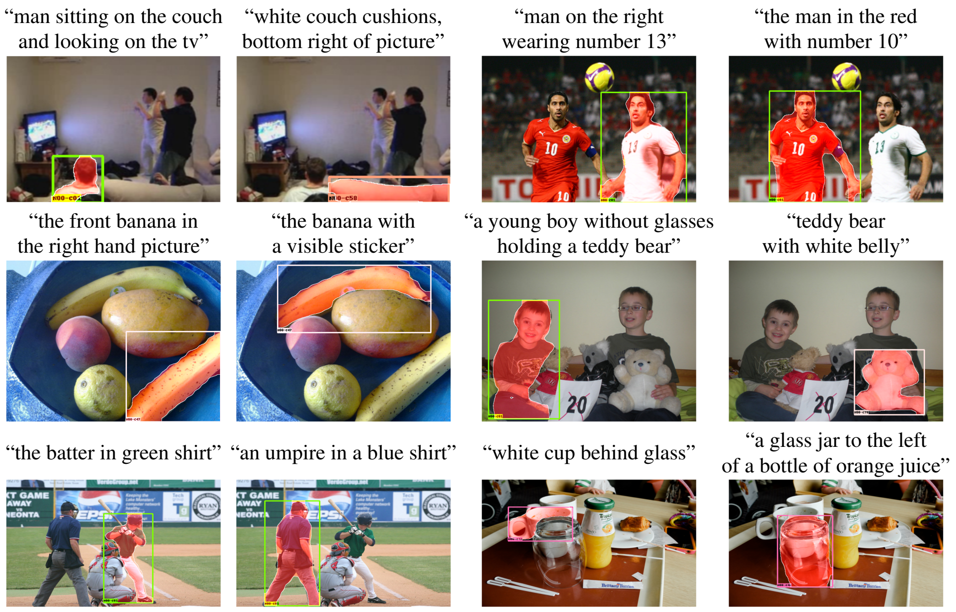
"man sitting on the couch and looking on the tv"
"white couch cushions, bottom right of picture"
"man on the right wearing number 13"
"the man in the red with number 10"
"the front banana in the right hand picture"
"the banana with a visible sticker"
"a young boy without glasses holding a teddy bear"
"teddy bear with white belly"
"the batter in green shirt"
"an umpire in a blue shirt"
"white cup behind glass"
"a glass jar to the left of a bottle of orange juice"

Figure 6: Sample results of objects referred by different query expressions on the RefCOCOg dataset.

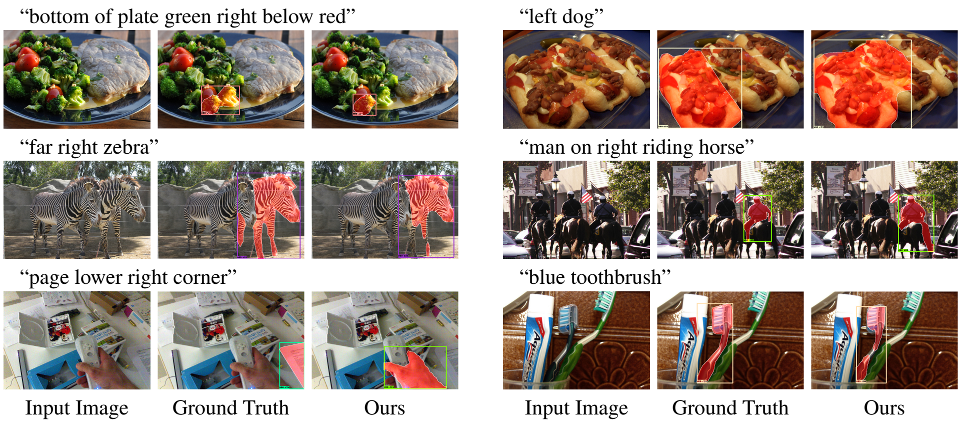

Figure 7: Failure cases of our method on the RefCOCO dataset.