Supplementary Material for
GRES: Generalized Referring Expression Segmentation

Chang Liu[†] Henghui Ding[†✉] Xudong Jiang
Nanyang Technological University
https://henghuiding.github.io/GRES

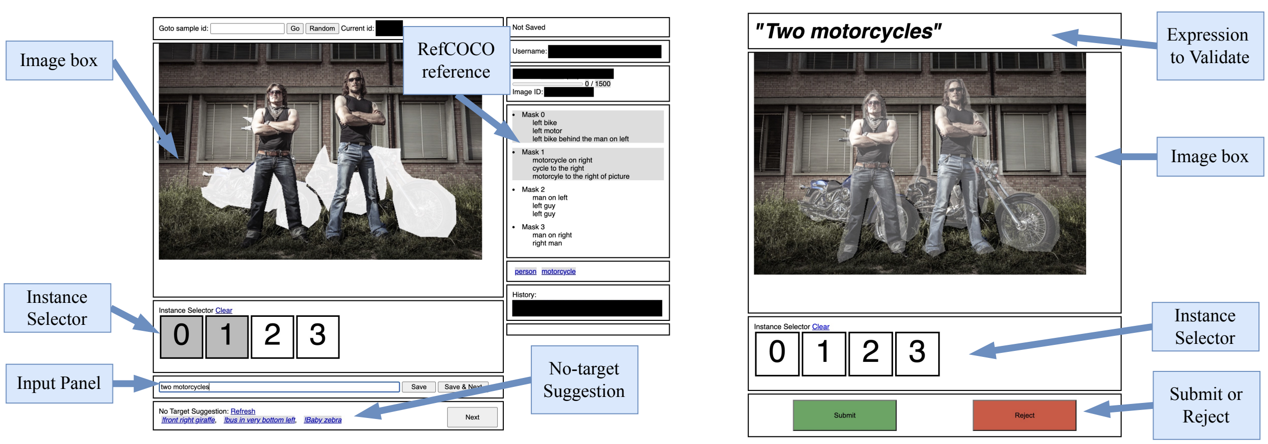
This supplementary material contains two parts: **1).** More details and examples of the proposed dataset gRefCOCO (Appendix A); **2).** More experimental results and implementation details (Appendix B).

A. More Details and Examples of the Proposed Dataset gRefCOCO

A.1. Dataset Partitioning

gRefCOCO follows the UNC splitting of RefCOCO [26] and have four non-overlapped sub-sets: *train*, *val*, *testA*, *testB*. The *train* set is a superset of the *train* set of RefCOCO, with new images from the training set of MSCOCO added. Images for validation and testing (*val*, *testA* and *testB*) are strictly identical with RefCOCO, to avoid the risk of data leakage.

A.2. Annotation Procedure and Tool



(a) Annotation tool (b) Validation tool

Figure I. The screenshots of the developed annotation system used for building gRefCOCO.

Following ReferIt [11], the gRefCOCO is constructed in a game-like interactive manner, in which annotations and validations are done alternatively by two players: one annotator and one validator. We developed a web-based annotation system to facilitate the annotation and validation work. The system contains two parts: an annotation tool and a validation tool. Screenshots are shown in Fig. I.

[†]Equal contribution.
✉ Corresponding author (henghui.ding@gmail.com).

Annotation. As shown in Fig. Ia, the annotation tool can randomly draw an image from the COCO dataset, load all object masks of this image, and display them in the Image Box. The annotator is required to select a set of targets from the image using the Instance Selector, and write the referring expression in the Input Panel. The annotator is allowed to check the RefCOCO's referring expressions of this image for reference if possible. Finally, after the annotator clicks the submit button, the annotated sample will be automatically sent to the validation side.

As we mentioned in Sec. 3.2 in the main paper, our system can generate no-target expression suggestions by randomly drawing expressions of other images in RefCOCO. Annotators can either write no-target expressions by themselves or select a deceptive expression from the suggestions. All suggested expressions are drawn from the same split as the current annotating split to avoid data leakage, *e.g.*, if the annotator is annotating the *train* set of gRefCOCO, all suggestions will come from the *train* set of RefCOCO.

Validation. Fig. Ib shows a screenshot of the validation tool. After the validation side receives a sample from the annotation side, it displays the sample's image and expression on the top of the page, then asks the validator to select and submit the targets referred by this expression. The annotator's selected targets will not be shown to the validator, so the validator needs to find targets independently. After the validator submits their selection, the backend system compares the targets found by the validator with the annotation submitted by the annotator. If they are identical, *i.e.*, the validator and the annotator independently selected the same targets, this sample is accepted as a valid gRefCOCO sample. Otherwise, this sample will be sent to another validator for a second check. Then if the second validator still fails to target this sample, it will be discarded. Validators can also directly reject samples that are inappropriate or do not meet the quality requirements. For no-target samples, the validator also needs to do a submission without instance selection to confirm. They are also required to reject no-target expressions that are totally irrelevant to the image.

A.3. More Examples of gRefCOCO

More samples of gRefCOCO are shown in Fig. III and Fig. II.

"the dog lying on left" *"oranges in the bowl"* *"the guy standing in back"* *"blueberry box"*

Figure II. Example no-target expressions of gRefCOCO.

B. More Experiments

B.1. Implementation Details

Our framework utilizes BERT-base-uncased [3] as language encoder. To achieve a fair comparison with previous works, single-target model utilizes Swin-base [15] backbone with feature fusing following previous work [23]. Images are resized to 480×480 before sending into the network. The BERT language model uses the default config of huggingface's implementation [21], and is frozen until the last two layers. The pixel decoder contains 5 Transformer decoder layers. The channel numbers of all hidden layers in the prediction head are set to 256. AdamW optimizer with a weight decay of 0.01 is used to train the whole network. Learning rate is set to 1e-5 at the beginning, and is decreased by 10 times at 10,000-th and 140,000-th iteration. The model is trained for 150,000 iterations with a batch size of 24 on four 32G V100 GPUs.

B.2. More Comparisons on Classic RES

In Tab. I, we report the comparison of our methods with more previous methods on classic RES. We achieve new state-of-the-art performance on three RES datasets consistently. Even compared with methods trained with extra image-text data,

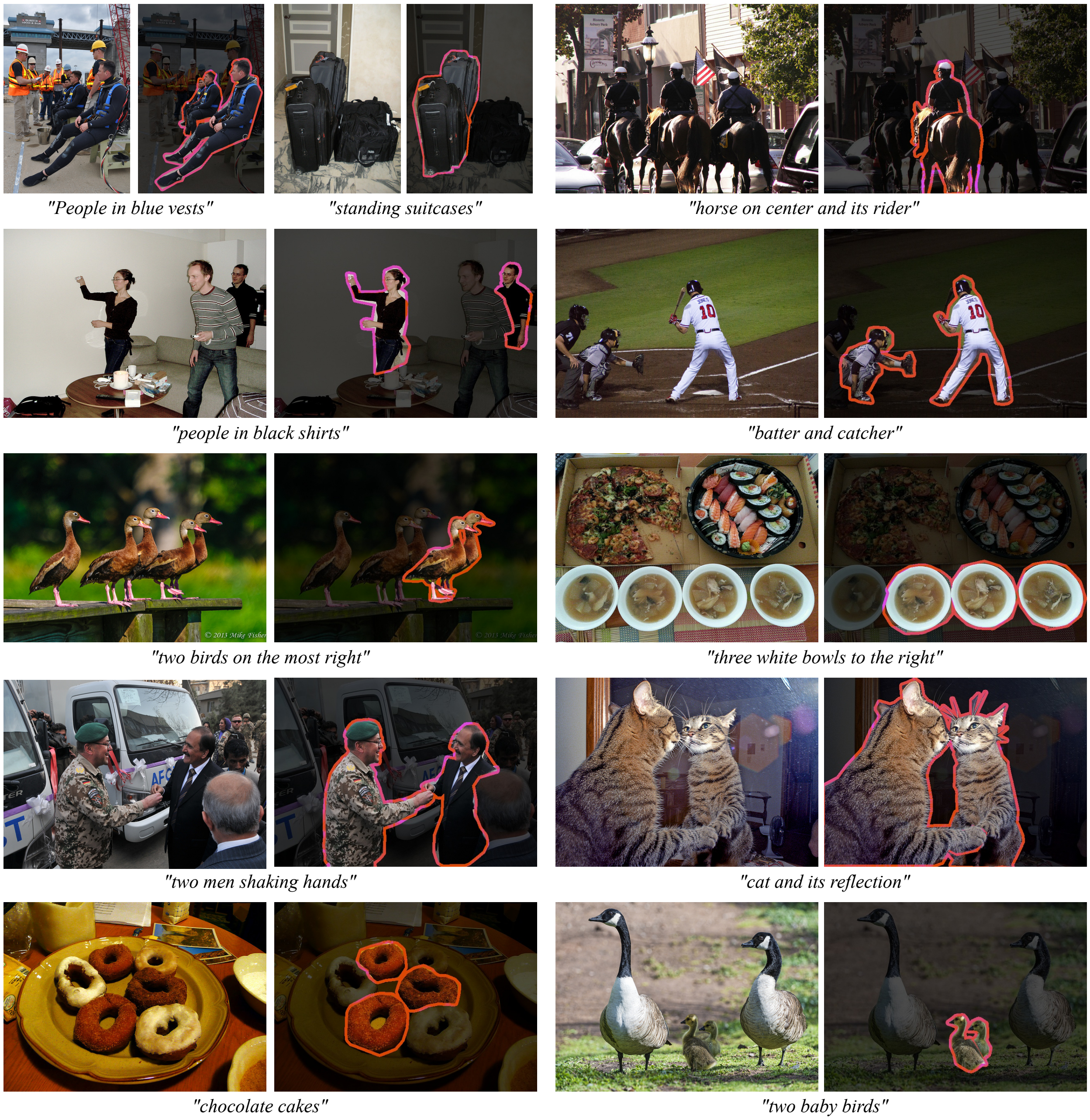


Figure III. Example multi-target expressions of gRefCOCO. Left: image; right: ground-truth.

e.g., CRIS [20] that adopts CLIP [19] trained on large-scale image-text datasets, our model still achieves better performance.

B.3. Fair Comparison of ReLA on Classic RES

To eliminate the influence of different visual/textual encoders, we compare our methods with other methods under the same visual encoder and textual encoder. In Tab. II, besides LAVT [23] and VLT [5] that originally have the same backbone as ours, we re-implement more classic RES methods: LTS [10] and EFN [6] using Swin-Base [15] as visual encoder and BERT [3] as textual encoder. We test these methods on the classic RES to give a fair comparison. All methods, including ours, are trained on the RefCOCO dataset only. As shown in Tab. II, all CNN-based methods get huge performance gains with the stronger transformer-based backbones. Especially for EFN [6], a performance boost of 8% can be achieved after

Table I. Results on classic RES in terms of cIoU. U: UMD split. G: Google split.

Methods	Visual Encoder	Textual Encoder	RefCOCO			RefCOCO+			G-Ref		
			val	test A	test B	val	test A	test B	val$_{(U)}$	test$_{(U)}$	val$_{(G)}$
DMN [18]	DPN92	SRU	49.78	54.83	45.13	38.88	44.22	32.29	-	-	36.76
RRN [13]	Deeplab-101	LSTM	55.33	57.26	53.93	39.75	42.15	36.11	-	-	36.45
MAttNet [25]	Res101-mrcn	LSTM	56.51	62.37	51.70	46.67	52.39	40.08	47.64	48.61	-
CMSA [24]	Deeplab-101	None	58.32	60.61	55.09	43.76	47.60	37.89	-	-	39.98
CAC [2]	ResNet101	LSTM	58.90	61.77	53.81	-	-	-	46.37	46.95	44.32
STEP [1]	Deeplab-101	LSTM	60.04	63.46	57.97	48.19	52.33	40.41	-	-	46.40
BRINet [7]	Deeplab-101	LSTM	60.98	62.99	59.21	48.17	52.32	42.11	-	-	48.04
CMPC [8]	Deeplab-101	LSTM	61.36	64.53	59.64	49.56	53.44	43.23	-	-	39.98
LSCM [9]	Deeplab-101	LSTM	61.47	64.99	59.55	49.34	53.12	43.50	-	-	48.05
MCN [17]	Darknet53	GRU	62.44	64.20	59.71	50.62	54.99	44.69	49.22	49.40	-
CMPC+ [14]	Deeplab-101	LSTM	62.47	65.08	60.82	50.25	54.04	43.47	-	-	49.89
EFN [6]	ResNet101	GRU	62.76	65.69	59.67	51.50	55.24	43.01	-	-	51.93
BUSNet [22]	Deeplab-101	Self-Att	63.27	66.41	61.39	51.76	56.87	44.13	-	-	50.56
CGAN [16]	Deeplab-101	GRU	64.86	68.04	62.07	51.03	55.51	44.06	51.01	51.69	46.54
LTS [10]	Darknet53	GRU	65.43	67.76	63.08	54.21	58.32	48.02	54.40	54.25	-
VLT [4]	Darknet53	GRU	67.52	70.47	65.24	56.30	60.98	50.08	54.96	57.73	52.02
ReSTR [12]	ViT-B	Transformer	67.22	69.30	64.45	55.78	60.44	48.27	-	-	54.48
CRIS [20]	CLIP	CLIP	70.47	73.18	66.10	62.27	68.08	53.68	59.87	60.36	-
LAVT [23]	Swin-B	BERT	72.73	75.82	68.79	62.14	68.38	55.10	61.24	62.09	60.50
VLT [5]	Swin-B	BERT	72.96	75.96	69.60	63.53	68.43	56.92	63.49	**66.22**	**62.80**
ReLA (ours)	Swin-B	BERT	**73.82**	**76.48**	**70.18**	**66.04**	**71.02**	**57.65**	**65.00**	65.97	62.70
ReLA (ours)$_{\text{mIoU}}$	Swin-B	BERT	75.61	77.79	72.82	70.42	74.83	63.87	68.65	69.56	66.89

Table II. Fair comparison with other methods with the same visual/textual encoders on val set of RefCOCO. †: re-implementation with Swin-B [15] & BERT [3].

Methods	Pr@0.5	Pr@0.6	Pr@0.7	Pr@0.8	Pr@0.9	IoU	mIoU
LTS† [10]	80.72	73.62	71.03	62.84	27.23	69.64	70.98
EFN† [6]	82.68	75.00	72.37	63.26	29.45	70.83	72.41
VLT† [5]	83.69	75.63	73.01	65.30	28.77	71.26	72.84
LAVT [23]	84.46	-	75.28	-	34.30	72.73	74.46
LAVT† [23]	84.69	76.82	75.82	66.58	34.56	72.63	74.74
ReLA (ours)	**85.92**	**83.02**	**77.71**	**68.10**	**34.99**	**73.82**	**75.61**

changing the backbone. Our method outperforms the previous state-of-the-art LAVT [23] by more than 1% IoU.

B.4. More Failure Cases and Analysis

Though our method outperforms other methods on GRES, some failure cases are worth noting. Figure IV shows more failure cases of our model. Sample (a) and (b) uses hard and rare descriptions, *e.g.* *"in front row"* and *"turned off"*, to refer to a set of targets. Such kind of expressions hardly appears in the single-target classic datasets. In image (c), two cups on the left are very close, but one cup is on the plate while the other is not. This requires future works to have the ability to distinguish such small details of objects. Sample (d) is a no-target sample. There does exist a lady pulling a suitcase, but the suitcase color in the expression is wrong. This suggests that models need to pay more attention to details in both image and the language expression.

Sample (e) is a case showing the challenging feature of GRES over RES. In this sample, the green frisbee is spatially closer to the center kid but is held by another kid on the left. Two success cases are generated by our method trained only on the RES dataset. It can be seen that the RES model successfully finds either of the target kids. However, in GRES, the network is confused about the center kid. This is because, in classic RES, the network only needs to output the most possible instance, so it does not need to care about the girl in the center. But in GRES, as the number of output instances is not arbitrary, the network also needs to judge whether each instance should be outputted, even if it is not the most possible one.

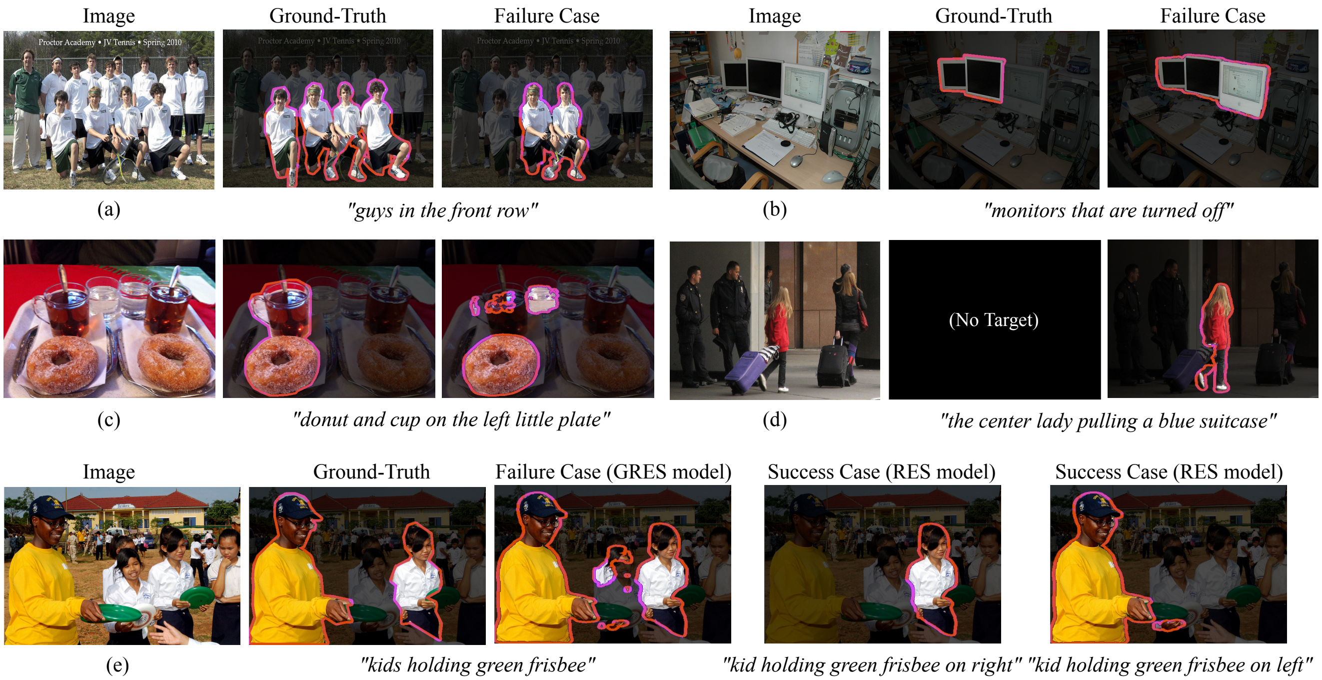


Figure IV. More failure cases of our method on the proposed dataset gRefCOCO.

References

[1] Ding-Jie Chen, Songhao Jia, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. See-through-text grouping for referring image segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 7454–7463, 2019. 4

[2] Y.-W. Chen, Y.-H. Tsai, T. Wang, Y.-Y. Lin, and M.-H. Yang. Referring expression object segmentation with caption-aware consistency. In *Proc. Brit. Mach. Vis. Conf.*, 2019. 4

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. NAACL-HLT*, volume 1, pages 4171–4186. Association for Computational Linguistics, 2019. 2, 3, 4

[4] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 16321–16330, 2021. 4

[5] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vlt: Vision-language transformer and query generation for referring segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 3, 4

[6] Guang Feng, Zhiwei Hu, Lihe Zhang, and Huchuan Lu. Encoder fusion network with co-attention embedding for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021. 3, 4

[7] Zhiwei Hu, Guang Feng, Jiayu Sun, Lihe Zhang, and Huchuan Lu. Bi-directional relationship inferring network for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 4424–4433, 2020. 4

[8] Shaofei Huang, Tianrui Hui, Si Liu, Guanbin Li, Yunchao Wei, Jizhong Han, Luoqi Liu, and Bo Li. Referring image segmentation via cross-modal progressive comprehension. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10488–10497, 2020. 4

[9] Tianrui Hui, Si Liu, Shaofei Huang, Guanbin Li, Sansi Yu, Faxi Zhang, and Jizhong Han. Linguistic structure guided context modeling for referring image segmentation. In *Proc. Eur. Conf. Comput. Vis.*, pages 59–75. Springer, 2020. 4

[10] Ya Jing, Tao Kong, Wei Wang, Liang Wang, Lei Li, and Tieniu Tan. Locate then segment: A strong pipeline for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 9858–9867, 2021. 3, 4

[11] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proc. of the Conf. on Empirical Methods in Natural Language Process.*, pages 787–798, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. 1

[12] Namyup Kim, Dongwon Kim, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Restr: Convolution-free referring image segmentation using transformers. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 18145–18154, 2022. 4

[13] Ruiyu Li, Kaican Li, Yi-Chun Kuo, Michelle Shu, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Referring image segmentation via recurrent refinement networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 5745–5753, 2018. 4

[14] Si Liu, Tianrui Hui, Shaofei Huang, Yunchao Wei, Bo Li, and Guanbin Li. Cross-modal progressive comprehension for referring segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(9):4761–4775, 2022. 4

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 10012–10022, 2021. 2, 3, 4

[16] Gen Luo, Yiyi Zhou, Rongrong Ji, Xiaoshuai Sun, Jinsong Su, Chia-Wen Lin, and Qi Tian. Cascade grouped attention network for referring expression segmentation. In *ACM Int. Conf. Multimedia*, pages 1274–1282, 2020. 4

[17] Gen Luo, Yiyi Zhou, Xiaoshuai Sun, Liujuan Cao, Chenglin Wu, Cheng Deng, and Rongrong Ji. Multi-task collaborative network for joint referring expression comprehension and segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10034–10043, 2020. 4

[18] Edgar Margffoy-Tuay, Juan C Pérez, Emilio Botero, and Pablo Arbeláez. Dynamic multimodal instance segmentation guided by natural language queries. In *Proc. Eur. Conf. Comput. Vis.*, pages 630–645, 2018. 4

[19] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[20] Zhaoqing Wang, Yu Lu, Qiang Li, Xunqiang Tao, Yandong Guo, Mingming Gong, and Tongliang Liu. Cris: Clip-driven referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 11686–11695, 2022. 3, 4

[21] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics. 2

[22] Sibei Yang, Meng Xia, Guanbin Li, Hong-Yu Zhou, and Yizhou Yu. Bottom-up shift and reasoning for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021. 4

[23] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 18155–18165, 2022. 2, 3, 4

[24] Linwei Ye, Mrigank Rochan, Zhi Liu, and Yang Wang. Cross-modal self-attention network for referring image segmentation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 10502–10511, 2019. 4

[25] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1307–1315, 2018. 4

[26] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Proc. Eur. Conf. Comput. Vis.*, pages 69–85. Springer, 2016. 1