GSVA: Generalized Segmentation via Multimodal Large Language Models Supplementary Materials

A. Discussions

A.1. Comparison to Concurrent Works

NExT-Chat [22] decodes boxes and masks from the ⟨trigger⟩ token, showing promising capabilities in many grounded understanding tasks. However, it falls short of explicitly rejecting non-existing objects in user's queries. **SESAME** [20] is adept at correcting the wrong referents and segmenting the closest object in the image by adjusting the input prompt with an alternative to the empty target, while GSVA tackles it with **[REJ] tokens in a unified output space**. When there are multiple empty targets, GSVA can seamlessly reject them, while SESAME could have undefined behaviors. In summary, the aforementioned concurrent works consider the challenge of multiple and empty targets in the area of segmentation LLMs to a certain extent, while GSVA studies the problem **more systematically via weight-sharing SEG tokens and the novel REJ token**.

A.2. GSVA and Reasoning Segmentation

We evaluate GSVA on ReasonSeg [11] with a generalized configuration. Table A1 demonstrates the capability of GSVA to handle instructions with complex logic, showing competitive results to LISA[1]. Figure A1 exemplifies that GSVA can segment the ReasonSeg referent while rejecting the additional empty target, while LISA fails to make the rejection, which further verifies the efficacy of GSVA.

Method	gIoU	cIoU
LISA-7B (ft)	50.5	53.2
GSVA-7B (ft)	50.5	56.4

Table A1. Results of LISA and GSVA on ReasonSeg dataset.

A.3. Support of Various Question Types

For simplicity, we show one type of question in the example in the paper. In contrast, the prompt types are diversified during training, including "Please segment {objs} in this image", "Can you segment {objs} in the image", *etc*. Besides, the training data also contains VQA and Reasoning Segmentation, which include various questions and answers. As shown in Table A2, apart from the substituted "Where" question of "What," we also examine the robustness of GSVA by testing the pretrained model with the unseen "Show" and "Outline" questions, short for "Show me {objs} in the image with segmentation masks" and "Outline {objs} in this image with segmentation masks," respectively. As demonstrated in Figure A2, different question forms do not impact much performance deviation as long as they are reasonable.

Question	gIoU	cIoU	N-acc.
"What"	63.32	61.70	56.44
"Where"	63.43	61.57	56.98
"Show"	63.05	61.35	56.48
"Outline"	63.16	61.51	56.27

Table A2. GRES results with different types of questions.

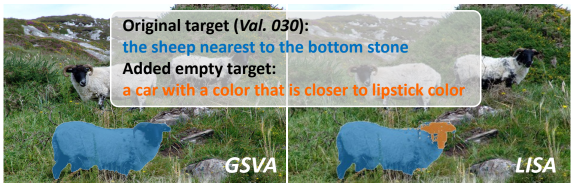

Original target (*Val. 030*):
the sheep nearest to the bottom stone
Added empty target:
a car with a color that is closer to lipstick color
GSVA
LISA

Figure A1. Generalized Reasoning Segmentation Example.

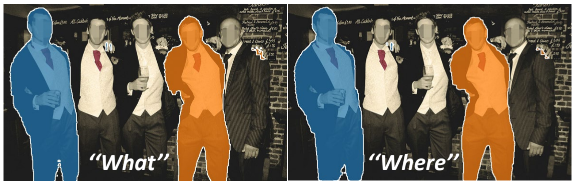


Figure A2. Example of different types of questions.

A.4. Multiple Objects in One Expression

In GSVA, each expression is separated with a comma in the prompt, corresponding to a segmentation map or rejection token. If more than one object is stated in the expression, a single [SEG] token will guide a segmentation mask to cover all objects. The common practice of GRES is merging all masks of the objects referred to into one as the ground truth for evaluation, which we follow for fair comparisons. Since there exist numerous expressions containing both present and absent objects, GSVA learns to predict the mask of the union of the referents so the absent objects will not occur.

A.5. Failure Cases on N-acc.

Although GSVA achieves a **state-of-the-art** level of N-acc on the GRES task and outperforms both Non-LLM and LLM models, we still observe that several failure cases are relatively small and vague fragments in the images, easily leading to misperception. As shown in Figure A3, there is

[1] Results reproduced by the open-source code of LISA.

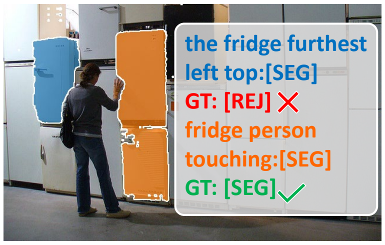

the fridge furthest left top:[SEG]
GT: [REJ] ✗
fridge person touching:[SEG]
GT: [SEG] ✓

Figure A3. Failure case example of GSVA in terms of N-acc., where empty targets are incorrectly predicted with [SEG] tokens.

no fridge obviously at the top left, but the model proposes a nearer fridge instead. It is hard for the model to tell whether in the corner stands a fridge, especially at low input resolution. This suggests using higher-resolution vision encoders, which will be an interesting future direction.

B. Implementation Details

We elaborate on the implementation details of GSVA and the settings of the experiments and list all the hyper-parameters in Table A3 for easy reproduction of the results.

Pretraining. In the pretraining stages, GSVA starts from a pretrained MLLM, *e.g.*, LLaVA-Vicuna-7B [13] and a pretrained SFM, *e.g.*, SAM-ViT-H [10]. As shown in Figure 2 in the main paper, the mask decoder F_{mask}, the segmentation query projector ψ, the LoRA [6] adapter weights on query and value projections in the LLM, and the token embeddings in the vocabularies are trainable. During training, the cross entropy language modeling loss on the output text sequence $\tilde{\mathbf{y}}_{\text{txt}}$ and the ground truth response $\mathbf{y}_{\text{txt}}$, the output mask $\tilde{\mathbf{y}}_{\text{mask}}$ is supervised by the combination of a binary cross entropy loss and a DICE loss to minimize the error to the ground truth mask $\mathbf{y}_{\text{mask}}$. To summarize, the final objective is

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \lambda_1 \mathcal{L}_{\text{LM}}(\tilde{\mathbf{y}}_{\text{txt}}, \mathbf{y}_{\text{txt}}) + \lambda_2 \mathcal{L}_{\text{BCE}}(\tilde{\mathbf{y}}_{\text{mask}}, \mathbf{y}_{\text{mask}}) \\ &+ \lambda_3 \mathcal{L}_{\text{DICE}}(\tilde{\mathbf{y}}_{\text{mask}}, \mathbf{y}_{\text{mask}}),\end{aligned} \tag{A1}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the weight of the above losses, following LISA [11] set to 1.0, 2.0, and 0.5, respectively.

In the pretraining phase, we mainly adopt the mixed dataset configurations in LISA [11] to mix four types of datasets and sample in training splits by the ratio 9 of Semantic Segmentation (SemSeg), 6 of Referring Expression Segmentation (RES), 3 of Visual Question Answering (VQA), and 1 of Reasoning Segmentation (ReasonSeg). This ratio of 9:6:3:1 is almost kept from LISA for a fair comparison, except for the increment of RES to weight gRefCOCO [12] more. For each task, the sample is uniformly sampled across the dataset. The semantic segmentation datasets consist of ADE20K [23], COCO-Stuff [1], Mapillary Vistas [17], PACO-LVIS [19], and PASCAL-Part [2]. The RES datasets include RefCOCO, RefCOCO+ [9], RefCOCOg [16], RefCLEF [9], and gRefCOCO [12]. VQA and reasoning segmentation adopt LLaVA-Instruct-150K [13] and ReasonSeg [11] as the datasets, respectively.

We train all three model variants (Vicuna-7B, Vicuna-13B, and Llama2-13B) on 8 Tesla A100 GPUs (80GB) for 50,000 steps with a batch size of 2 per device. The learning rate is set to 3×10^{-4} with a gradient accumulation of 10. The weight decay is set to 0, and the gradients are clipped to 1 by the maximum norm. The learning rate is warmed up in 100 steps and linearly decayed. To reduce the GPU memory footprints, we adopt the AdamW [14] optimizer with stage 2 ZeRO [18]. As for LoRA, we pick 8 as the LoRA rank for the 7B variant and 64 for the 13B variants.

Finetuning on GRES task. We finetune the pretrained models on gRefCOCO [12] dataset to improve the GRES performances further. For all variants of GSVA, we load the weights of the respective model in pretraining and train another 10 epochs on gRefCOCO. This is slightly different from pretraining because we go through the whole training set once per epoch without sampling data from it. In the finetuning experiments, we use a lower learning rate 1×10^{-4}, keeping other configurations unchanged.

Finetuning on classic RES task. For classic RES, including RefCOCO, RefCOCO+ [9], and RefCOCOg [16], we choose the mixed classic RES dataset during training aside from gRefCOCO. In addition to reducing the learning rate to 1×10^{-4}, the other hyper-parameters remain unchanged.

C. GSVA on Ref-ZOM

Ref-ZOM dataset. Proposed by DMMI [7], Ref-ZOM is a similar dataset to gRefCOCO [12], posing the challenges of one-to-one, one-to-many, and one-to-zero referring expression to targets in the image relationships in RES. The one-to-one is the case of classic RES where the referring expression is matched with only one target in the image, while the one-to-many and one-to-zero are the multi-target and empty-target cases, respectively. Ref-ZOM contains 55,075 images and 74,942 annotated objects, among which there are 56,972 one-to-one cases, 21,290 one-to-many cases, and 11,937 one-to-zero cases. DMMI provides a default split of training set and test set of Ref-ZOM. There are 43,749 images in the training set and 11,329 images in the test set.

Setups. We regard Ref-ZOM [7] as a kind of GRES, and use a similar protocol to gRefCOCO [12] to evaluate LISA and GSVA on Ref-ZOM dataset. For Ref-ZOM, gIoU and cIoU are substituted to the equivalent metrics, mIoU and oIoU. Different from gRefCOCO, mIoU, and oIoU only count for non-empty targets. For one-to-zero, *i.e.*, empty targets, the accuracy is one if the predicted mask is strictly all-zero. LISA [11] and our proposed GSVA with Vicuna-7B [3] are evaluated, including the pretrained versions and the finetuned versions with 1 epoch on the training split.

Results. As shown in Table A4, without finetuning on Ref-

Experiment	Configuration	Model		
		GSVA-Vicuna-7B	GSVA-Vicuna-13B	GSVA-Llama2-13B
Pretraining	Dataset Types	SemSeg, RES, VQA, ReasonSeg		
	SemSeg Datasets	ADE20K [23], COCO-Stuff [1], Maplilary Vistas [17], PACO-LVIS [19], Pascal-Part [2]		
	RES Datasets	RefCOCO [9], RefCOCO+ [9], RefCOCOg [16], RefCLEF [9], gRefCOCO [12]		
	VQA Datasets	LLaVA-Instruct-150K [13]		
	ReasonSeg Datasets	ReasonSeg [11]		
	Epochs / Steps	50,000 steps, gradient accumulation: 10 steps / update		
	Optimizer	AdamW [14], learning rate: 3×10^{-4}, weight decay: 0.0, gradient clip: 1.0		
	ZeRO [18]	Stage: 2		
	Batch size	2 samples / GPU×8 GPUs		
	LoRA [6] rank	8	64	64
Finetuning on gRefCOCO [12]	Dataset Types	RES		
	SemSeg Datasets	-		
	RES Datasets	gRefCOCO [12]		
	VQA Datasets	-		
	ReasonSeg Datasets	-		
	Epochs / Steps	10 epochs, gradient accumulation: 10 steps / update		
	Optimizer	AdamW [14], learning rate: 1×10^{-4}, weight decay: 0.0, gradient clip: 1.0		
	ZeRO [18]	Stage: 2		
	Batch size	2 samples / GPU×8 GPUs		
	LoRA [6] rank	8	64	64
Finetuning on classic RES	Dataset Types	RES		
	SemSeg Datasets	-		
	RES Datasets	RefCOCO [9], RefCOCO+ [9], RefCOCOg [16], RefCLEF [9]		
	VQA Datasets	-		
	ReasonSeg Datasets	-		
	Epochs / Steps	50,000 steps, gradient accumulation: 10 steps / update		
	Optimizer	AdamW [14], learning rate: 1×10^{-4}, weight decay: 0.0, gradient clip: 1.0		
	ZeRO [18]	Stage: 2		
	Batch size	2 samples / GPU×8 GPUs		
	LoRA [6] rank	8	64	64

Table A3. Detailed configurations and hyper-parameters of GSVA in pretraining and finetuning stages. "steps" mean the model is trained for given steps, which is implemented by fixing the steps in each epoch and sample data from the datasets, while "epochs" mean the model is trained across the whole dataset in each epoch.

ZOM, GSVA surpasses LISA by clear margins in oIoU and mIoU of over 6%. Also, GSVA approaches close to the SOTA, DMMI [7], which is the SOTA proposed along with the Ref-ZOM dataset. After finetuning for 1 epoch, LISA quickly catches up GSVA with less 2% IoU metrics, while GSVA achieves 94.59% accuracy in classifying empty targets, keeping a competitive performance to DMMI.

Method	Ref-ZOM [7] Test Set		
	oIoU	mIoU	Acc
MCN [15]	55.03	54.70	75.81
CMPC [8]	56.19	55.72	77.01
VLT [4]	60.21	60.43	79.26
LAVT [21]	64.45	64.78	83.11
DMMI [7]	**68.77**	**68.21**	87.02
LISA-Vicuna-7B [11]	60.14	61.46	72.58
GSVA-Vicuna-7B	67.12	67.98	82.66
LISA-Vicuna-7B [11] (ft)	66.41	65.39	93.39
GSVA-Vicuna-7B (ft)	68.29	68.13	**94.59**

Table A4. GRES results on the test split of Ref-ZOM [7] dataset. oIoU and mIoU are only computed on the samples containing targets, while the correct prediction of Acc is the mask of all zeros for empty targets. Baselines are excerpted from Hu et al. [7].

D. GSVA on Semantic Segmentation

To verify the vanilla semantic segmentation ability, we include an extra evaluation of the pretrained GSVA-Vicuna-7B and LISA-Vicuna-7B on the ADE20K [23] validation dataset. Since the pretraining covers the training set of ADE20K, we only evaluate them to see if they are capable of segmenting semantic regions rather than some specific objects. We prompt the models with the instruction as ***User:*** *What is* $\{classname\}$ *in this image?* ***Assistant:*** *Sure, [SEG].*, where each $\{classname\}$ is filled with the class name that exists in this image. We report the mIoU of the predictions and the ground truths, in Table A5. LISA achieves 60.11 mIoU whereas our GSVA slightly improves to 60.56.

Model	mIoU(ss)
LISA-Vicuna-7B [11]	60.11
GSVA-Vicuna-7B	60.56

Table A5. Semantic Segmentation Results of GSVA and LISA [11] on ADE20K validation dataset. The mIoU is different from the one in the closed-set segmentation, which is computed with the existing classes, and resembles the recall to some extent.

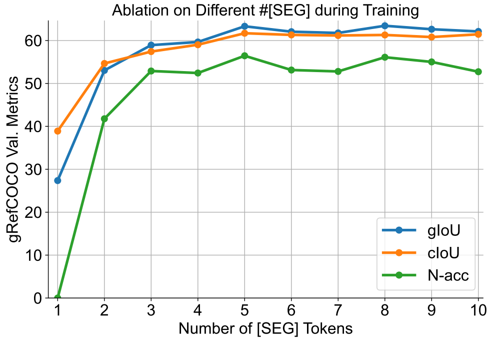


Figure A4. Ablation on how many [SEG] tokens are used during training. Metrics including gIoU, cIoU, and N-acc on gRefCOCO [12] dataset are reported.

E. More Ablation Study

The number of targets in training. We study the effect on how many [SEG] tokens are adopted in training GSVA. In GSVA, the number of [SEG]s controls the number of targets processed by GSVA, which impacts the final capacity of multiple target segmentation. We sweep this hyperparameter from 1 to 10 and report the results of GSVA-Vicuna-7B on gRefCOCO [12] validation set in Figure A4. On the one hand, from the figure, all the metrics include gIoU, cIoU, and N-acc. increases as the number of [SEG] grows from 1 to 5, and the model performances saturate after 5 [SEG] tokens are involved. On the other hand, the memory footprints keep rising as more [SEG] tokens are used and exceed 24GB (RTX3090 / RTX4090) after 6 [SEG] tokens. Considering the performance saturation and the memory consumption, we pick 5 as the default.

Weight sharing for different [SEG] tokens. Another design choice considering multiple [SEG] tokens is to use a set of individual [SEG] embeddings rather than sharing weights in GSVA. We adopt 8 tokens from [SEG000] to [SEG007] to study this design. In case of more than 8 targets, we dispatch the targets based on the remainder after divided by 8. However, this design poses convergence difficulties, as shown in Table A6. We attribute the intensive performance degradation to the slow convergence of the [SEG] token embedding, which is that too many [SEG] token embeddings could interfere with each other. Therefore, it is reasonable to share weights between [SEG] tokens.

Model	gRefCOCO Val.		
	gIoU	cIoU	N-acc.
GSVA (weight sharing)	63.32	61.70	56.45
w/ 8 individual embeddings	16.13	23.11	0.00

Table A6. Ablation study on weight sharing of [SEG] tokens.

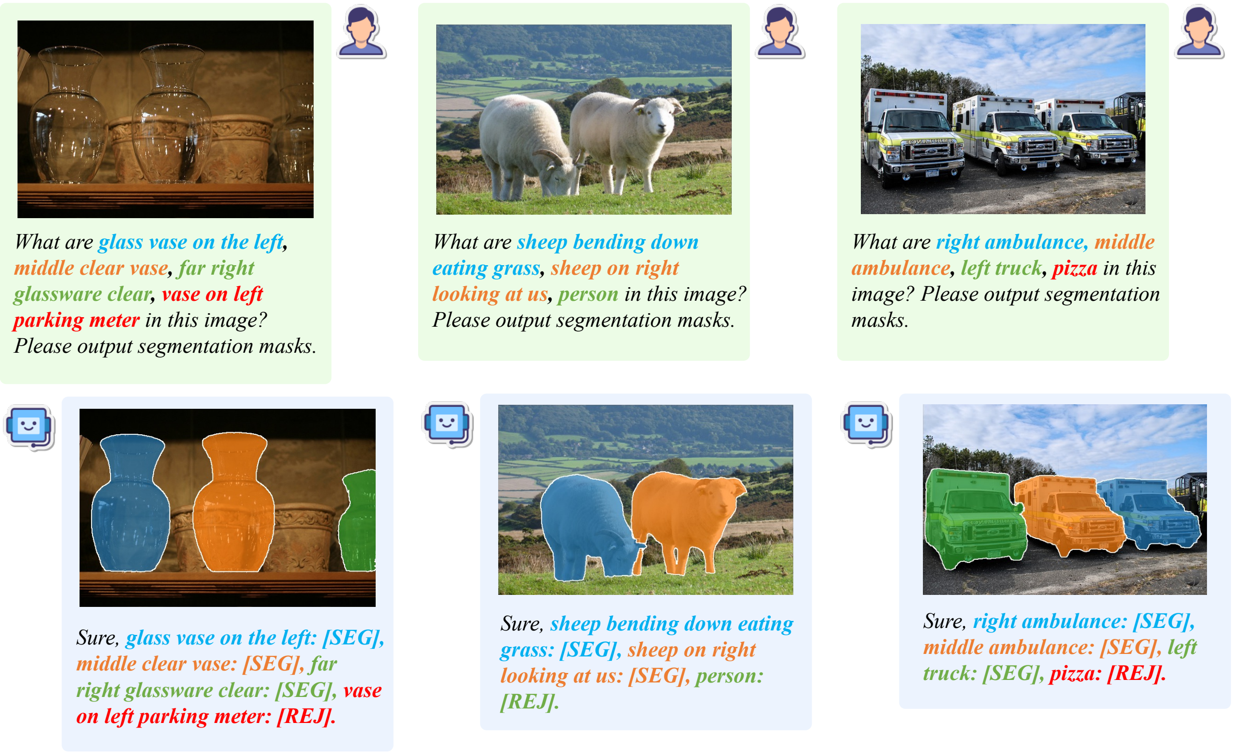
Vision Encoder F_{V2} in SFM	gRefCOCO Val.		
	gIoU	cIoU	N-acc.
SAM-ViT-H (GSVA)	63.32	61.70	56.45
SAM-ViT-L	61.67	60.46	56.04
SAM-ViT-B	59.20	56.36	58.57

Table A7. Ablation study on Different SAM [10] ViT backbones.

Different SAM Backbones. We alter the default pretrained SAM-ViT-H [10] vision encoder to SAM-ViT-L and SAM-ViT-B, to verify the effectiveness of GSVA that is stable to the change of segmentation vision encoder F_{V2}. As shown in Table A7, when we substitute the ViT-H [5] backbones with SAM pretrained ViT-B/L, the segmentation performances decreases by 4%∼5% while the accuracy of empty targets change mildly and even increases to 58.57% with ViT-B. The slight drops in gIoU and cIoU scores could be attributed to the usages of more miniature vision encoders, and the stable, even higher N-acc implies that generating [REJ] token by MLLM consistently works well with SFMs in different capacities. These results also demonstrate the effectiveness of the design that rejects the empty targets by a special token rather than predicting an all-zero mask.

F. More Visualizations

We provide more visualization results in Figure A5 to further show the effectiveness of our GSVA in GRES task.



*What are **glass vase on the left**, **middle clear vase**, **far right glassware clear**, **vase on left parking meter** in this image? Please output segmentation masks.*

*Sure, **glass vase on the left: [SEG]**, **middle clear vase: [SEG]**, **far right glassware clear: [SEG]**, **vase on left parking meter: [REJ]**.*

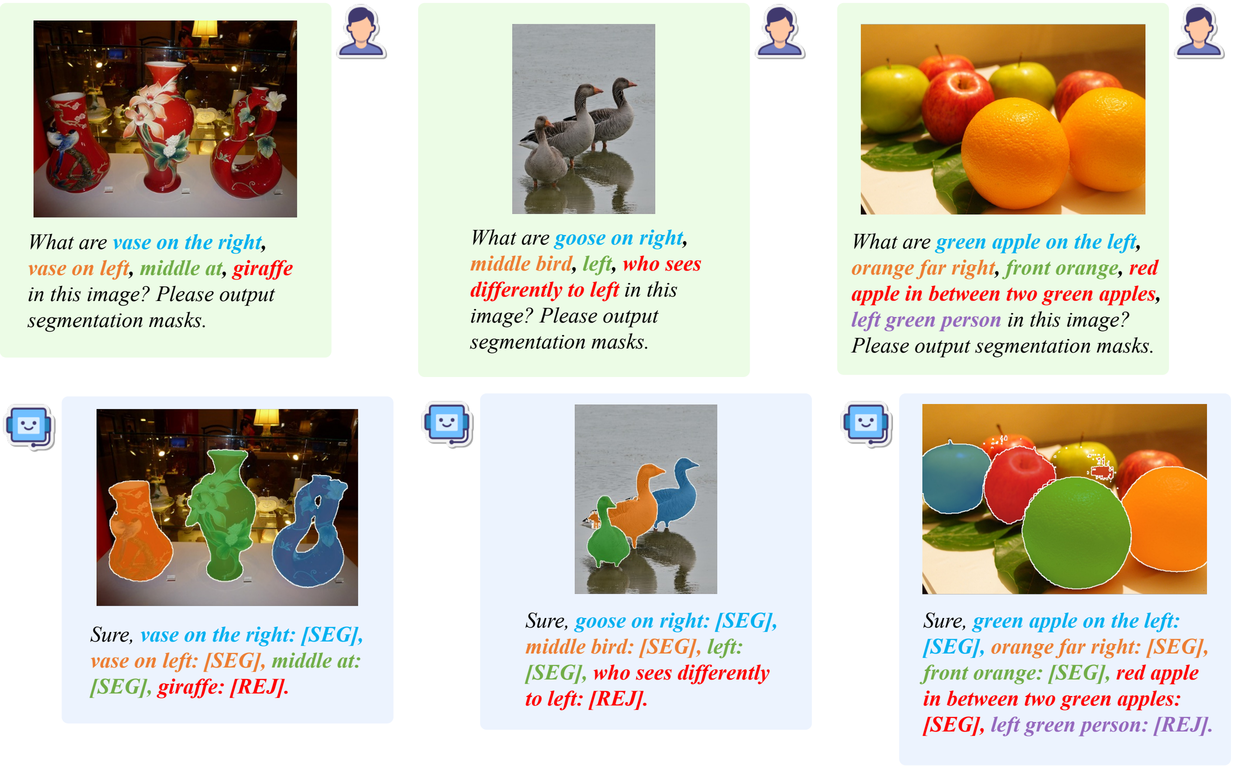
*What are **sheep bending down eating grass**, **sheep on right looking at us**, **person** in this image? Please output segmentation masks.*

*Sure, **sheep bending down eating grass: [SEG]**, **sheep on right looking at us: [SEG]**, **person: [REJ]**.*

*What are **right ambulance**, **middle ambulance**, **left truck**, **pizza** in this image? Please output segmentation masks.*

*Sure, **right ambulance: [SEG]**, **middle ambulance: [SEG]**, **left truck: [SEG]**, **pizza: [REJ]**.*

(a) Examples of GSVA on gRefCOCO [12] test set A and B. (I)



*What are **vase on the right**, **vase on left**, **middle at**, **giraffe** in this image? Please output segmentation masks.*

*Sure, **vase on the right: [SEG]**, **vase on left: [SEG]**, **middle at: [SEG]**, **giraffe: [REJ]**.*

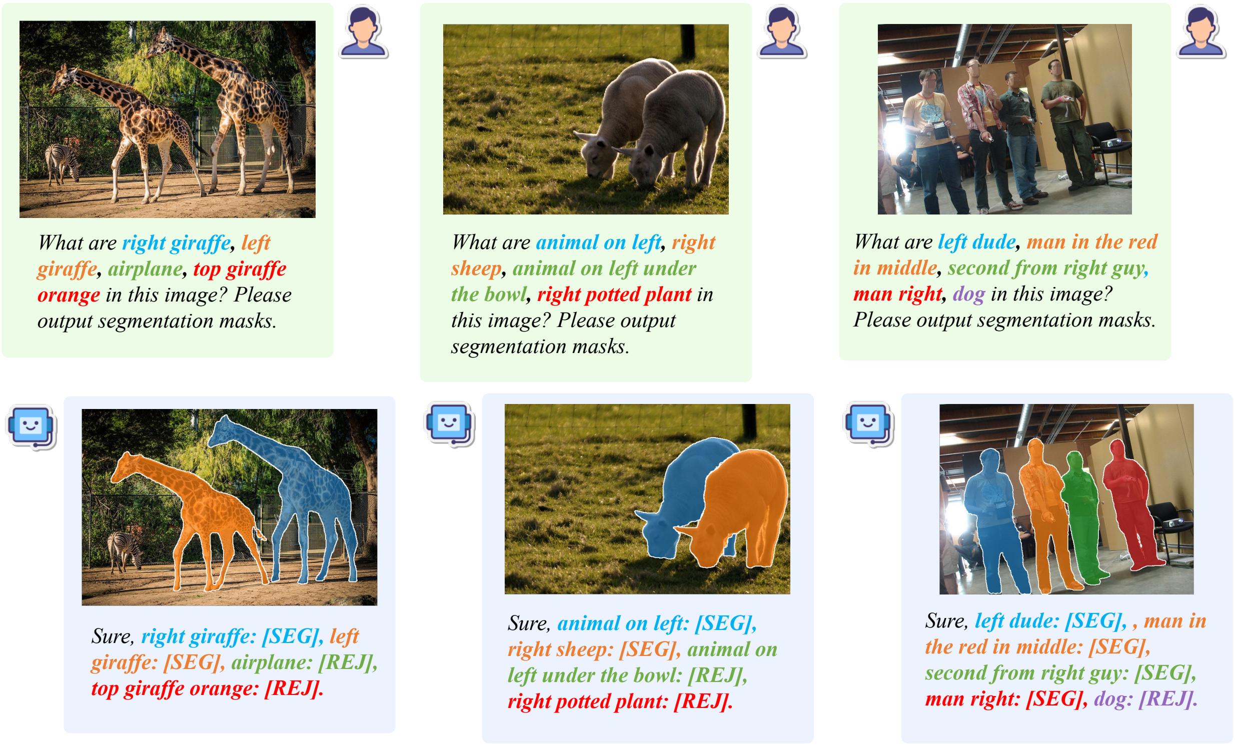
*What are **goose on right**, **middle bird**, **left**, **who sees differently to left** in this image? Please output segmentation masks.*

*Sure, **goose on right: [SEG]**, **middle bird: [SEG]**, **left: [SEG]**, **who sees differently to left: [REJ]**.*

*What are **green apple on the left**, **orange far right**, **front orange**, **red apple in between two green apples**, **left green person** in this image? Please output segmentation masks.*

*Sure, **green apple on the left: [SEG]**, **orange far right: [SEG]**, **front orange: [SEG]**, **red apple in between two green apples: [SEG]**, **left green person: [REJ]**.*

(b) Examples of GSVA on gRefCOCO [12] test set A and B. (II)



What are ***right giraffe****,* ***left giraffe****,* ***airplane****,* ***top giraffe orange*** *in this image? Please output segmentation masks.*

Sure, ***right giraffe: [SEG], left giraffe: [SEG], airplane: [REJ], top giraffe orange: [REJ].***

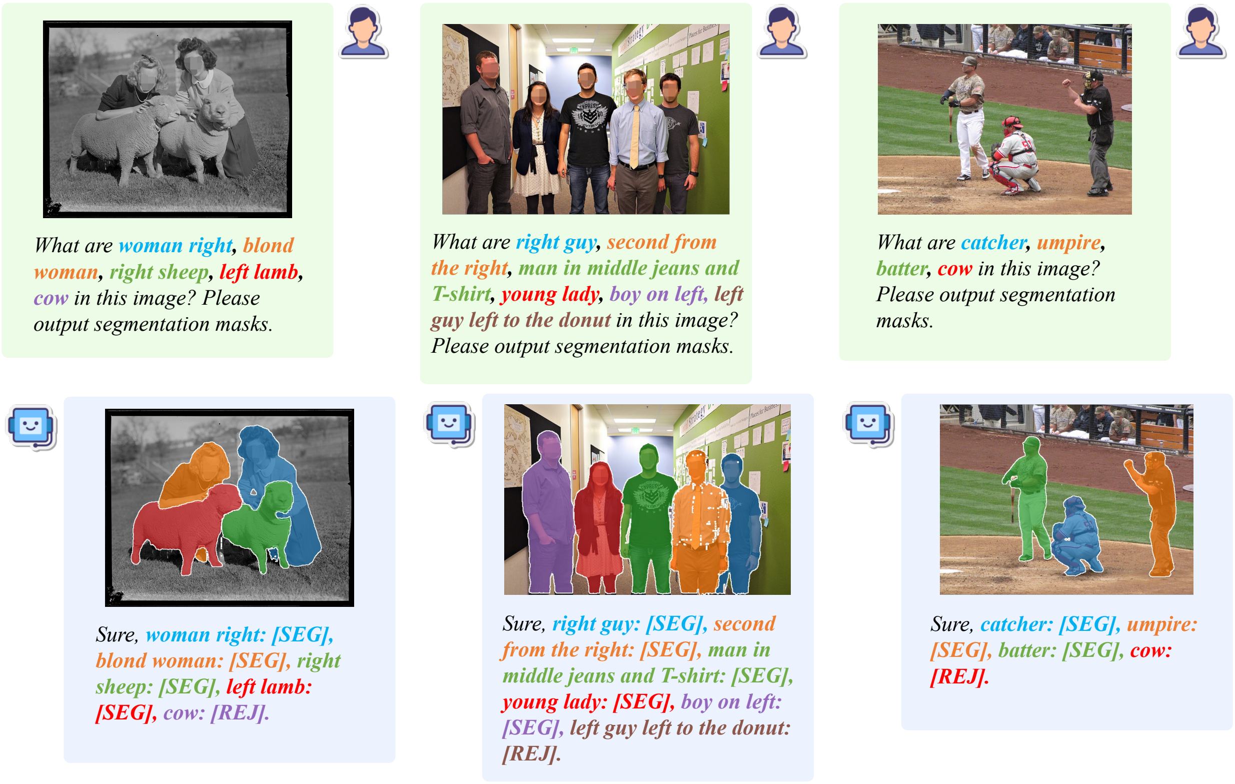
What are ***animal on left****,* ***right sheep****,* ***animal on left under the bowl****,* ***right potted plant*** *in this image? Please output segmentation masks.*

Sure, ***animal on left: [SEG], right sheep: [SEG], animal on left under the bowl: [REJ], right potted plant: [REJ].***

What are ***left dude****,* ***man in the red in middle****,* ***second from right guy****,* ***man right****,* ***dog*** *in this image? Please output segmentation masks.*

Sure, ***left dude: [SEG], , man in the red in middle: [SEG], second from right guy: [SEG], man right: [SEG], dog: [REJ].***

(c) Examples of GSVA on gRefCOCO [12] test set A and B. (III)



What are ***woman right****,* ***blond woman****,* ***right sheep****,* ***left lamb****,* ***cow*** *in this image? Please output segmentation masks.*

Sure, ***woman right: [SEG], blond woman: [SEG], right sheep: [SEG], left lamb: [SEG], cow: [REJ].***

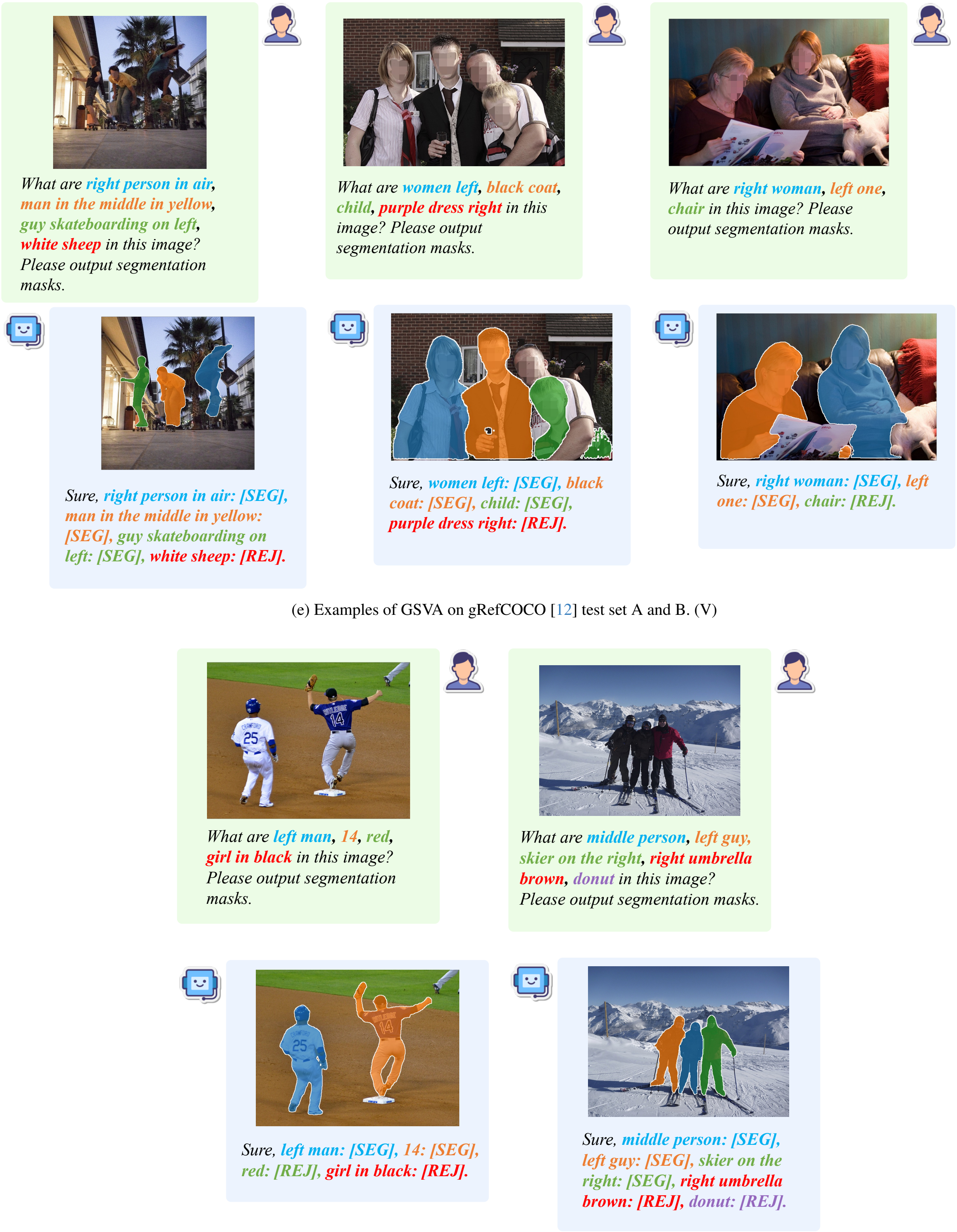
What are ***right guy****,* ***second from the right****,* ***man in middle jeans and T-shirt****,* ***young lady****,* ***boy on left****,* ***left guy left to the donut*** *in this image? Please output segmentation masks.*

Sure, ***right guy: [SEG], second from the right: [SEG], man in middle jeans and T-shirt: [SEG], young lady: [SEG], boy on left: [SEG], left guy left to the donut: [REJ].***

What are ***catcher****,* ***umpire****,* ***batter****,* ***cow*** *in this image? Please output segmentation masks.*

Sure, ***catcher: [SEG], umpire: [SEG], batter: [SEG], cow: [REJ].***

(d) Examples of GSVA on gRefCOCO [12] test set A and B. (IV)



*What are **right person in air**, **man in the middle in yellow**, **guy skateboarding on left**, **white sheep** in this image? Please output segmentation masks.*

*Sure, **right person in air: [SEG]**, **man in the middle in yellow: [SEG]**, **guy skateboarding on left: [SEG]**, **white sheep: [REJ].***

*What are **women left**, **black coat**, **child**, **purple dress right** in this image? Please output segmentation masks.*

*Sure, **women left: [SEG]**, **black coat: [SEG]**, **child: [SEG]**, **purple dress right: [REJ].***

*What are **right woman**, **left one**, **chair** in this image? Please output segmentation masks.*

*Sure, **right woman: [SEG]**, **left one: [SEG]**, **chair: [REJ].***

(e) Examples of GSVA on gRefCOCO [12] test set A and B. (V)

*What are **left man**, **14**, **red**, **girl in black** in this image? Please output segmentation masks.*

*Sure, **left man: [SEG]**, **14: [SEG]**, **red: [REJ]**, **girl in black: [REJ].***

*What are **middle person**, **left guy**, **skier on the right**, **right umbrella brown**, **donut** in this image? Please output segmentation masks.*

*Sure, **middle person: [SEG]**, **left guy: [SEG]**, **skier on the right: [SEG]**, **right umbrella brown: [REJ]**, **donut: [REJ].***

(f) Examples of GSVA on gRefCOCO [12] test set A and B. (VI)

Figure A5. Visualization of GSVA in GRES task, the inputs and outputs are presented in the form of dialogues between human user and the chatbot. The examples are selected from gRefCOCO [12] test set A and B. Zoom in for best view.

References

[1] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In *IEEE CVPR*, 2018. 2, 3

[2] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *IEEE CVPR*, 2014. 2, 3

[3] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2

[4] Henghui Ding, Chang Liu, Suchen Wang, and Xudong Jiang. Vision-language transformer and query generation for referring segmentation. In *IEEE ICCV*, 2021. 3

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4

[6] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 2, 3

[7] Yutao Hu, Qixiong Wang, Wenqi Shao, Enze Xie, Zhenguo Li, Jungong Han, and Ping Luo. Beyond one-to-one: Rethinking the referring image segmentation. In *IEEE ICCV*, 2023. 2, 3

[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 *IEEE CVPR*, 2020. 3

[9] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 2, 3

[10] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *IEEE ICCV*, 2023. 2, 4

[11] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 1, 2, 3, 4

[12] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *IEEE CVPR*, 2023. 2, 3, 4, 5, 6, 7

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2, 3

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2, 3

[15] 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 *IEEE CVPR*, 2020. 3

[16] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *IEEE CVPR*, 2016. 2, 3

[17] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE ICCV*, 2017. 2, 3

[18] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *IEEE SC*, 2020. 2, 3

[19] Vignesh Ramanathan, Anmol Kalia, Vladan Petrovic, Yi Wen, Baixue Zheng, Baishan Guo, Rui Wang, Aaron Marquez, Rama Kovvuri, Abhishek Kadian, et al. Paco: Parts and attributes of common objects. In *IEEE CVPR*, 2023. 2, 3

[20] Tsung-Han Wu, Giscard Biamby, David Chan, Lisa Dunlap, Ritwik Gupta, Xudong Wang, Joseph E Gonzalez, and Trevor Darrell. See, say, and segment: Teaching lmms to overcome false premises. *arXiv preprint arXiv:2312.08366*, 2023. 1

[21] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *IEEE CVPR*, 2022. 3

[22] Ao Zhang, Liming Zhao, Chen-Wei Xie, Yun Zheng, Wei Ji, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*, 2023. 1

[23] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2019. 2, 3