Latent Expression Generation for Referring Image Segmentation and Grounding
- *Supplementary materials* -

Table of Contents

We provide a table of contents for the supplementary:

A. Dataset Details

A.1. RefCOCO

RefCOCO [19] is a dataset for referring image segmentation (RIS) and referring expression comprehension (REC), built on images and annotations (segmentation masks and bounding boxes) from MS-COCO [7]. It was collected using the approach used in ReferItGame [5], where one player writes a referring expression for a segmented object, and another player selects the corresponding object in an image. The dataset contains 142,210 expressions for 50,000 objects across 19,994 images. It is divided into train, validation, and test sets. The test set is further split into testA and testB. The testA contains images with people, while the testB includes images with all other objects. This split structure allows for separate evaluation of human and non-human referents.

A.2. RefCOCO+

RefCOCO+ [19] follows the same approach as RefCOCO but prohibits spatial terms like "left" or "right" and provides expressions based on object attributes. Thus, RefCOCO+ is characterized as a more challenging dataset than RefCOCO, as it requires accurate object localization using only visual attributes without relying on positional cues. It includes 141,564 expressions for 49,856 objects across 19,992 images, and is split in the same way as RefCOCO.

A.3. RefCOCOg

RefCOCOg [11, 12], unlike RefCOCO and RefCOCO+, was collected in a non-interactive setting via Amazon Mechanical Turk (AMT), resulting in the longer and more detailed textual forms. While RefCOCO and RefCOCO+ have concise expressions with an average length of 3.61 and 3.53 words, respectively, RefCOCOg expressions are significantly longer, averaging 8.43 words. This dataset was designed to evaluate a model's ability to comprehend more complex and contextually rich referring expressions. RefCOCOg consists of 95,010 expressions for 49,822 objects across 25,799 images and is divided into two partitions: the *Google* [11] split and the *UMD* [12] split. In the *Google* split, objects are separately assigned to either the train or validation set while allowing the same image to appear in both sets without object overlaps. The *UMD* split separates the data into train, validation, and test sets.

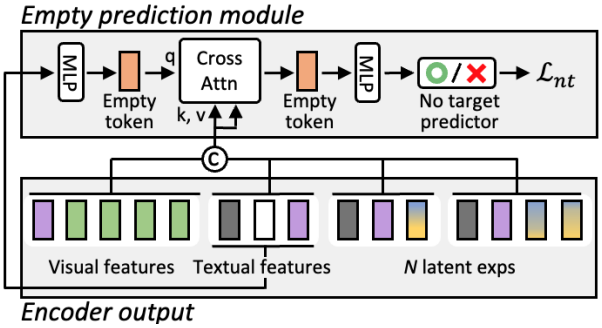

Figure 5. The illustration of our GRES framework, where the empty prediction module is added to the original framework after the feature extraction of the encoder. We also remove the subject distributor within an encoder. Other processes except for these minimal modifications are the same as the original one.

Methods	Encoders		RefCOCO			RefCOCO+			RefCOCOg		
	Visual	Textual	val	testA	testB	val	testA	testB	val(U)	test(U)	val(G)
*Trained on each **Single RefCOCO Dataset***											
Dual-encoder based Methods											
VPD[23] [21]	VQGAN	CLIP	73.46	75.31	70.23	61.41	67.98	54.99	63.12	63.59	-
CGFormer[23] [14]	Swin-B	BERT	74.75	77.30	70.64	64.54	71.00	57.14	64.68	64.09	62.51
RISCLIP[24] [6]	CLIP-B	CLIP	73.57	76.46	69.76	65.53	70.61	55.49	64.10	65.09	-
ReMamber[24] [18]	VMamba-B	CLIP	74.54	76.74	70.89	65.00	70.78	57.53	63.90	64.00	-
Single-encoder based Methods											
Shared-RIS[24] [20]	BEiT3-B		75.50	76.66	73.03	70.34	73.75	**65.07**	68.50	69.17	66.65
One-Ref[24] [17]	BEiT3-B		**77.55**	**80.96**	73.51	70.82	74.53	64.06	70.68	70.61	-
Latent-VG (ours)	BEiT3-B		77.41	79.92	**74.83**	**70.92**	**74.56**	63.68	**70.74**	**70.82**	**69.19**
*Trained on **Combined RefCOCO Dataset***											
Dual-encoder based Methods											
PolyFormer[23] [9]	Swin-B	BERT	74.82	76.64	71.06	67.64	72.89	59.33	67.76	69.05	-
ReMamber[24] [18]	VMamba-B	CLIP	75.06	78.27	71.82	64.40	69.49	56.34	66.70	68.05	-
SAM based Methods											
Chen. et al[24] [1]	SAM + Swin-B	BERT	75.37	77.20	71.38	68.07	73.46	59.47	67.75	69.50	-
Prompt-RIS[24] [13]	SAM + CLIP-B	CLIP	76.36	80.37	72.29	67.06	73.58	58.96	64.79	67.16	69.01
LLM based Methods											
GSVA-7B[24] [16]	SAM + CLIP-L	Vicuna	77.13	78.82	73.45	65.87	69.47	59.55	72.72	73.36	-
LaSagnA-7B[24] [15]	SAM + CLIP-L	Vicuna	76.30	77.38	72.76	64.42	67.62	58.63	71.13	72.01	-
Single-encoder based Methods											
One-Ref[24] [17]	BEiT3-B		**81.06**	**83.05**	77.80	72.24	77.32	67.08	75.14	**77.21**	-
Latent-VG (ours)	BEiT3-B		81.04	82.67	**79.77**	**75.27**	**78.25**	**69.65**	**75.88**	76.55	**75.02**

Table 8. oIoU comparison with other **RIS** methods on RefCOCO, RefCOCO+, and RefCOCOg datasets.

A.4. GRefCOCO

GRefCOCO is a dataset designed for Generalized Referring Expression Segmentation (GRES) [8], extending the standard RefCOCO dataset by supporting the cases of multi-target and no-target expressions. Unlike traditional RIS datasets, where each expression corresponds to a single existent object, GRefCOCO offers a description referring to multiple or non-existent objects. This makes the dataset more flexible and suited for real-world scenarios. The dataset consists of 278,232 referring expressions, including 80,022 multi-target expressions and 32,202 no-target expressions, covering 60,287 instances in 19,994 images.

B. Details on GRES Framework

B.1. Framework

In Fig. 5, we present the GRES framework, where we add the empty prediction modules to the original model by (1) introducing an empty token to handle no-target cases, (2) interacting the empty token with the output of the encoder via cross-attention, and (3) imposing a binary classification loss ($\mathcal{L}_{nt}$) on the empty prediction. We also remove a subject distributor within an encoder. All other processes are the same as the original framework.

B.2. Implementation Details for GRES

The differences in the detailed implementations for a GRES framework lie in (1) halving the learning rate from 0.0001 to 0.00005 for more stable training on complex GRES scenarios, and (2) applying the loss weight of 0.5 to the empty binary classification objective ($\mathcal{L}_{nt}$). Other training recipes are identical to the original visual grounding framework.

C. Additional Experiments

C.1. oIoU Results for RIS.

Tab. 8 shows the oIoU performance of our Latent-VG compared to SoTA RIS methods on the RIS benchmarks. In the single dataset setting, we achieve superior performance over the methods [17, 20] based on the same backbone (*i.e.*, BEiT3-B) as ours. In the combined dataset setting, our Latent-VG surpass Prompt-RIS [13] and LaSagnA-7B [15], even though they employ larger backbones (e.g., SAM and CLIP-L) than ours.

C.2. Efficiency Comparison.

In Tab. 9, we compare the efficiency of our methods with other visual grounding methods. The methods [2–4, 6, 10, 14] utilizing multi-scale features in their decoder incur high FLOPs, whereas our simple decoding yields lower FLOPs over them with the best results. Although One-Ref [17] for a REC model is more efficient than ours, it does not simultaneously perform a RIS task.

C.3. Efficiency of Proposed Components.

In Tab. 10, we analyze the incremental computational cost introduced by each proposed module. Adapting the latent

RIS Methods				
Methods	Params	GFLOPs	mIoU	oIoU
DMMI [3]	341M	392	67.51	63.98
CGFormer [14]	252M	949	68.56	64.54
RISCLIP† [6]	375M	1380	69.16	65.53
Shared-RIS† [20]	239M	155	70.34	68.42
One-Ref† [17]	267M	-	71.25	70.82
Latent-VG (ours)	267M	198	**73.19**	**70.92**
REC Methods				
Methods	Params	GFLOPs	Acc.	
TransVG++†‡ [2]	206M	396	75.39	
MDETR‡ [4]	185M	642	81.13	
Grounding-DINO‡ [10]	342M	464	82.75	
One-Ref†‡ [17]	234M	162	86.38	
Latent-VG (ours)	267M	198	**86.41**	

Table 9. Efficiency and performance comparison with other RIS and REC methods on the validation set of RefCOCO+. † denotes that the public code is not released. ‡ indicates the models trained on additional grounding data (*e.g.*, Flickr30k or ReferIt) than ours.

expression initializer increases the computational cost by 11M parameters and 2 GFLOPs, due to (1) length transform layers $\{\phi_i\}_{i=1}^{N}$ for initializing latent attributes and (2) additional MLP heads for processing each class token during prediction. In contrast, adding the subject distributor incurs only a negligible increase in computational cost, with the additional parameters and GFLOPs being significantly lower than those of other modules. Incorporating the visual concept injector adds 1M parameters and 1 GFLOPs since the parameters of the concept tokens and FLOPs for handling concept tokens are introduced. The computation required for the loss function applied between class tokens is also negligible.

Methods	SD	VCI	Params	GFLOPs
No Latent Exps			255M	195
+ Latent Exps	✓		266M	197
		✓	266M	197
	✓	✓	267M	198
+ $\mathcal{L}_{\text{pos-cont}}$	✓	✓	267M	198

Table 10. Analysis of the computational cost in different components. *No Latent Exps* means the base model without any proposed methods. *SD* and *VCI* denote the subject distributor and the visual concept injector, respectively.

D. Analysis on Latent Expressions

D.1. Qualitative Analysis of each Expression

In Fig. 6, we visualize the prediction result of each latent expression as well as the averaged final prediction with diverse cases. Each prediction of individual expression (*i.e.*, *Input Exp.*, *Latent Exp.1* and *Latent Exp.2*) is obtained by thresholding the corresponding probability map before averaging them for the final prediction. In the first and second rows, we present the cases where the noisy output of the *Input Exp.* is complemented by the more precise output from the *Latent Exps*, resulting in a correct final prediction. In the third row, all predictions of each expression are noisy, yet the final averaged output is accurate. In the last row, we visualize a result that all outputs are generated precisely.

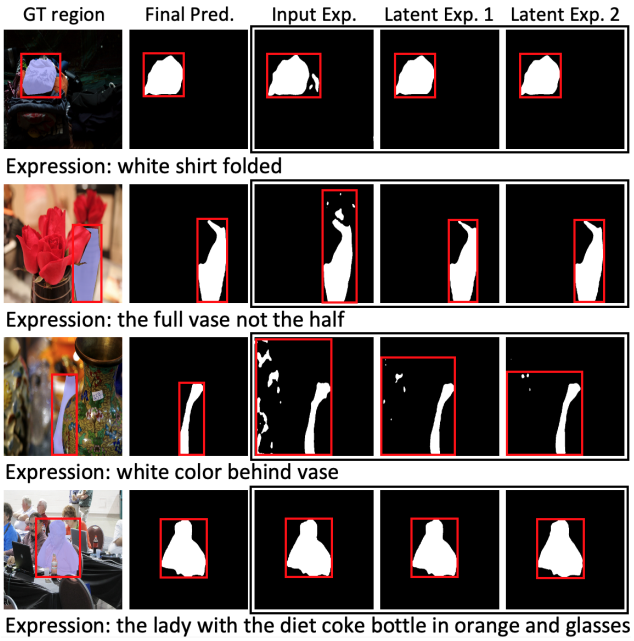

Figure 6. Qualitative analysis of each expression. The outputs for each expression (*i.e.*, *Input Exp.*, *Latent Exp.1* and *Latent Exp.2*) are obtained by thresholding the corresponding probability map before averaging them for the final prediction.

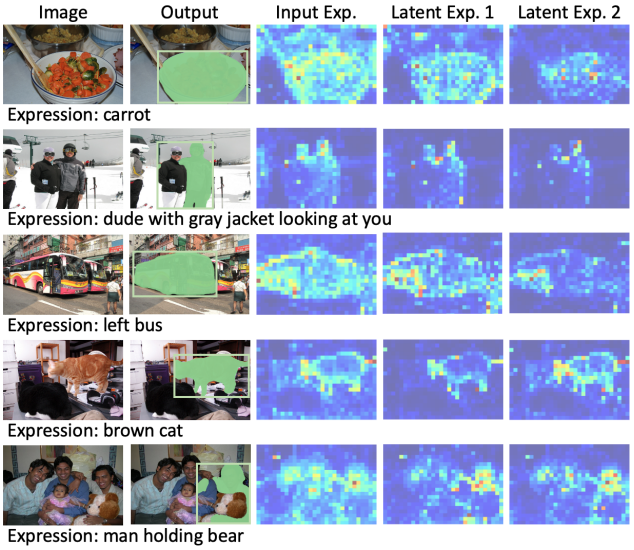

Figure 7. More attention maps on each expression.

D.2. More Attention Maps on each Expression

Fig. 7 presents additional attention maps for each expression. To generate these maps, we average the attention scores from all self-attention layers within an encoder. The variability in the locations of the maximum attention weight, along with the differently activated regions, indicates that each expression exhibits distinct visual details.

D.3. More Qualitative Analysis

In Fig. 8, we provide additional qualitative analysis of our Latent-VG compared to the base model without any of the proposed modules (termed as *No Latent Exp.*). The *No Latent Exp.* model often fails to distinguish the target from other similar objects when the limited textual cues are given. For instance, in the first row of Fig. 8, with the description "*donut with a hole nearest coffee*", the *No Latent Exp.* model captures a non-targeted donut, while our approach precisely selects the targeted donut, indicating our superiority in capturing the target cues.

D.4. IoU Scores for each Expression

In Tab. 11, we report the IoU scores on the validation of RefCOCO+ for each expression, as well as the final prediction. Each IoU score is calculated between the ground truth mask and the mask predicted by each expression. As all expressions are optimized by the identical loss function, they exhibit similar performances. However, the *Latent Exps* consistently achieve slightly higher IoU scores than the *Input Exp.*, even though they are derived from the input expression. This demonstrates the enhanced ability of the latent expressions to deliver target details into the model.

Metric	Input Exp.	Latent Exp.1	Latent Exp.2	Final Pred.
mIoU	72.84	**73.15**	73.07	73.19
oIoU	70.34	70.60	**70.63**	70.68

Table 11. IoU scores for each expression and the final prediction. Each IoU score is calculated between the ground truth mask and the mask predicted by each expression.

D.5. Convergence of each IoU Score.

Fig. 9 shows the convergence of IoU scores for each expression and the final prediction on the validation set of RefCOCO+. In the early stages of training, individual expressions exhibit varied IoU scores; however, after approximately 250 iterations, all scores converge to similar values. This convergence occurs because all expressions are optimized using the same learning objectives and are aligned similarly via the proposed contrastive loss.

D.6. Examples of an Extracted Subject

In Sec 3.1 of the main manuscript, we extract a subject token from the input textual tokens by applying a linear layer followed by a Gumble-softmax operation and an argmax function. The linear layer that generates subject logits is trained end-to-end using the framework learning objectives, without explicit supervision for the subject. Tab. 12 presents examples of the extracted subject tokens. In many cases, the correct subject is successfully extracted, probably because positioning the extracted token at the beginning of the latent expressions encourages accurate selection. However, in some failure cases, the extracted token does not correspond exactly to the true subject but instead captures a crucial keyword distinguishing the target (*e.g.*, for "*elephant in back*", the token "*back*" is selected). Moreover, when the true subject consists of multiple words (*i.e.*, mobile phone), only a single token (*e.g.*, mobile) is extracted, which may not fully represent the subject. We plan to explore these limitations

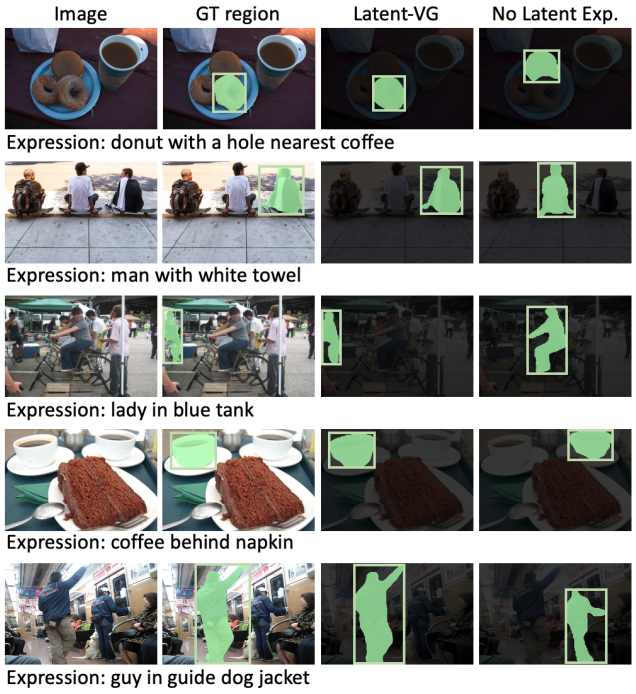

Figure 8. More qualitative analysis of Latent-VG compared to a model without any proposed methods (termed as *No Latent Exp.*).

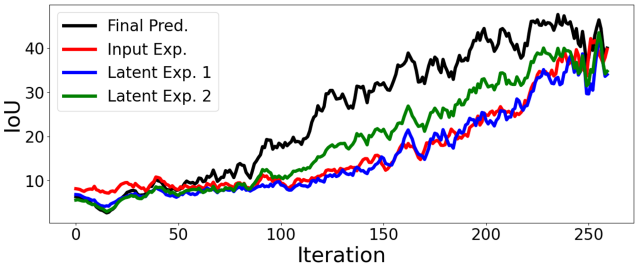

Figure 9. The convergence of IoU scores for each expression and the final prediction.

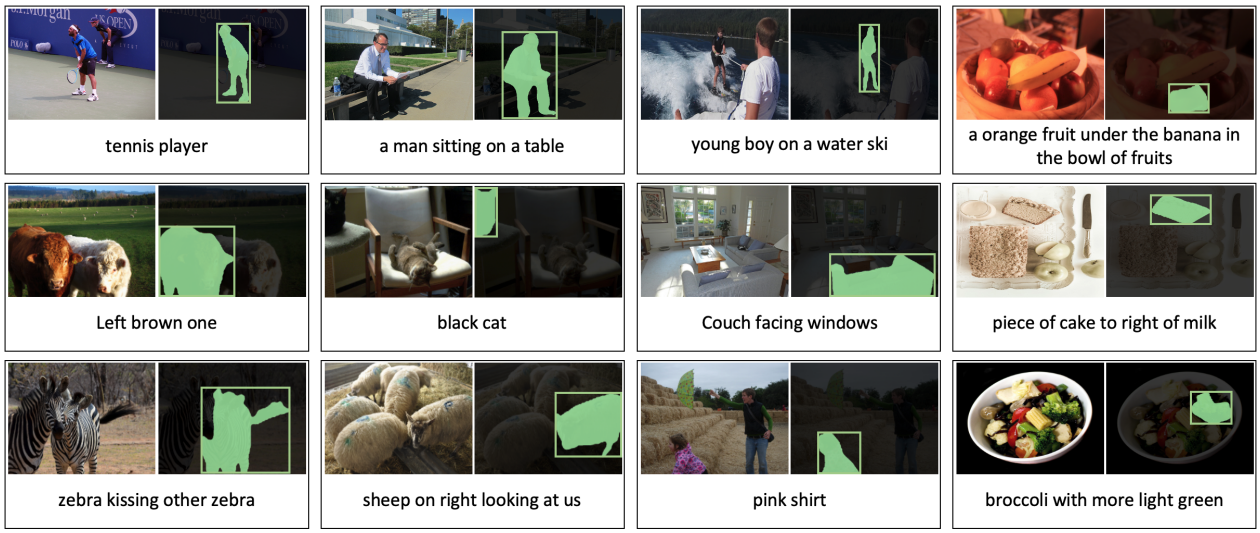

Figure 10. The visualization of segmentation and detection outputs of the proposed Latent-VG.

Correct Examples:
gray cat
bundle of broccoli
sugar powdered donut
sprinkle even with face almost
the zebra on the left in the right hand picture
a small girl starring at something along with her elder sister
a glass with napkins and utensils inside of it sitting near a pizza
Failure Examples:
elephant in back
the man 's hat
the vehicle on the left of the row
the mobile phone with a number 2125 towards the top right side

Table 12. Examples of the extracted subject in the input sentence.

in the future.

E. Further Discussion

E.1. Limitations

We discuss several limitations in our Latent-VG below:

The Reliance on the Input Expression. Our latent expressions are initialized from the input textual tokens, leading to the inevitable dependence on the input textual semantics. To mitigate this, we introduce to the dropout in Section 3.1 and select target-related patches as broad regions beyond the target area in the Visual Concept Injector from Sec. 3.2. in the main manuscript. Despite these efforts, our framework remains sensitive to the semantics of the input text. For example, as shown in Fig. 11, when identifying a suitcase with a downed zipper, the model must distinguish subtle differences in zipper locations among four suitcases. These extremely limited input cues lead our model to an incorrect suitcase. We tried to capture novel semantics outside of the input semantics in the latent representation, but if the input semantics occupy too small a portion of the target visual area, our model could select non-targeted objects.

Figure 11. The failure example of our Latent-VG.

The Weakness on the Small Size of Object. Since our method does not explicitly address the object scale variations, it performs less effectively on small-scale objects, as illustrated in Fig. 12. To assess this, we analyze the IoU scores based on the object size ratio (*i.e.* target object size divided by total image size). Our results reveal that performance drops significantly for object ratios in $0\% - 5\%$ and $5\% - 10\%$, showing a limitation in localizing small objects. We plan to investigate this issue further in future work.

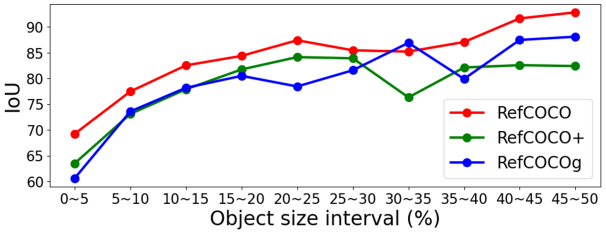

Figure 12. The IoU scores of our methods as the object size ratio.

The Incorrect Subject Selection. As discussed in Sec. D.6 and shown by the failure examples in Tab. 12, our

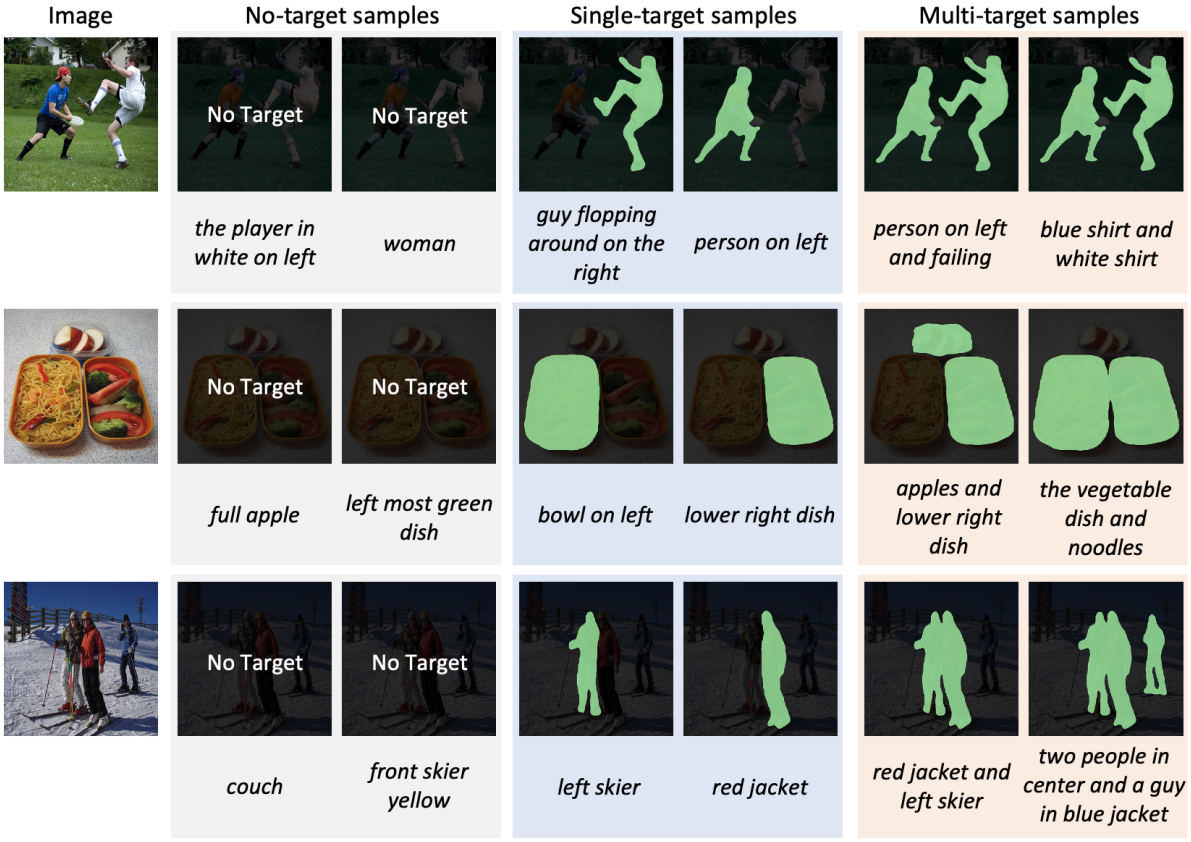

Figure 13. The visualization of predicted results of our Latent-VG for the GRES task.

design of subject selection can occasionally extract an incorrect subject. Although the incorrectly chosen token may sometimes capture a key distinguishing word (*e.g.*, "*left*" in "*the vehicle on the left of the row*", as reported in Tab. 12), this is not our intended outcome. We will explore alternative strategies for precise subject extraction, such as representing a subject token by operating a weighted sum over all textual tokens or incorporating subject supervision obtained via natural language processing (NLP) tools.

Despite these limitations, our methods achieve state-of-the-art performances in RIS, REC, and GRES tasks, demonstrating the effectiveness of our approach in leveraging latent expressions to capture novel semantics outside the input text.

E.2. Social Impact

Our work may inadvertently propagate biases present in the training data, leading to unintended ethical concerns. In addition, the capability to generate highly specified segmentation by users could be exploited for misinformation or deceptive media manipulation, further highlighting the importance of careful monitoring and regulation.

F. Visualizations

F.1. Visualization on RIS and REC

In Fig. 10, we present the visualizations of the outputs generated by our Latent-VG for the referring image segmentation (RIS) and referring expression comprehension (REC) on the RefCOCO(+/g) datasets.

F.2. Visualization on GRES

In Fig. 13, we visualize the examples of prediction results of our Latent-VG on the GRefCOCO dataset. Our framework demonstrates a strong ability to handle different referring descriptions, and the no-, single-, and multi-target scenarios for the same input images.

References

[1] Haoyuan Chen, Sihang Zhou, Kuan Li, Jianping Yin, and Jian Huang. A hybrid framework for referring image segmentation: Dual-decoder model with sam complementation. *Mathematics*, 12(19):3061, 2024. 2

[2] Jiajun Deng, Zhengyuan Yang, Daqing Liu, Tianlang Chen, Wengang Zhou, Yanyong Zhang, Houqiang Li, and Wanli Ouyang. TransVG++: End-to-end visual grounding with language conditioned vision transformer. *TPAMI*, 2023. 2, 3

[3] 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 *ICCV*, 2023. 3

[4] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. MDETR-modulated detection for end-to-end multi-modal understanding. In *ICCV*, 2021. 2, 3

[5] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 1

[6] Seoyeon Kim, Minguk Kang, Dongwon Kim, Jaesik Park, and Suha Kwak. Extending CLIP's image-text alignment to referring image segmentation. In *NAACL*, 2024. 2, 3

[7] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1

[8] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized Referring Expression Segmentation. In *CVPR*, 2023. 2

[9] Jiang Liu, Hui Ding, Zhaowei Cai, Yuting Zhang, Ravi Kumar Satzoda, Vijay Mahadevan, and R. Manmatha. PolyFormer: Referring Image Segmentation As Sequential Polygon Generation. In *CVPR*, 2023. 2

[10] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2, 3

[11] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 1

[12] Varun K Nagaraja, Vlad I Morariu, and Larry S Davis. Modeling context between objects for referring expression understanding. In *ECCV*, 2016. 1

[13] Chao Shang, Zichen Song, Heqian Qiu, Lanxiao Wang, Fanman Meng, and Hongliang Li. Prompt-driven referring image segmentation with instance contrasting. In *CVPR*, 2024. 2

[14] Jiajin Tang, Ge Zheng, Cheng Shi, and Sibei Yang. Contrastive Grouping With Transformer for Referring Image Segmentation. In *CVPR*, 2023. 2, 3

[15] Cong Wei, Haoxian Tan, Yujie Zhong, Yujiu Yang, and Lin Ma. LaSagnA: Language-based segmentation assistant for complex queries. *arXiv preprint arXiv:2404.08506*, 2024. 2

[16] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized segmentation via multimodal large language models. In *CVPR*, 2024. 2

[17] Linhui Xiao, Xiaoshan Yang, Fang Peng, Yaowei Wang, and Changsheng Xu. OneRef: Unified one-tower expression grounding and segmentation with mask referring modeling. In *NeurIPS*, 2024. 2, 3

[18] Yuhuan Yang, Chaofan Ma, Jiangchao Yao, Zhun Zhong, Ya Zhang, and Yanfeng Wang. Remamber: Referring image segmentation with mamba twister. In *ECCV*, 2024. 2

[19] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *ECCV*, 2016. 1

[20] Seonghoon Yu, Ilchae Jung, Byeongju Han, Taeoh Kim, Yunho Kim, Dongyoon Wee, and Jeany Son. A simple baseline with single-encoder for referring image segmentation. *arXiv preprint arXiv:2408.15521*, 2024. 2, 3

[21] Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *ICCV*, 2023. 2