Better, Stronger, Faster: Tackling the Trilemma in MLLM-based Segmentation with Simultaneous Textual Mask Prediction

Supplementary Material

In the supplementary materials, we report:

- **Implementation Details.** We provide comprehensive implementation details, including the training parameters and datasets for each experimental stage, the methodology for constructing instruction-following data, and a comparison of datasets used by our method versus baseline models (Sec. S1).
- **Additional Experiments.** We report detailed analyses of our model's efficiency, including inference time, GPU memory consumption (Sec. S2).
- **More Qualitative Showcases.** We present detailed qualitative showcases of *STAMP*, covering standard referring segmentation(in the main text), reasoning segmentation, visual question answering and advanced capabilities such as muti-round dialogue, multi-round segmentation and unified dialogue-segmentation examples (Sec. S3).
- **Code and Documentation.** We provide the source code for training and execution, along with a user guide and setup instructions (Sec. S4).

S1. Implementation Details

In this section, we detail our training strategies, data construction methodology, and the experimental setup for a fair comparison against baseline methods.

S1.1. Training Strategy

The specific training hyperparameters for the experiments in each section are detailed in Tab. S1, aligning with standard practices. Our experiments utilize the Qwen2-VL-2B and Qwen2-VL-7B models [S1]. We maintain consistent hyperparameters across experiments wherever possible to isolate the performance gains from our proposed methods, rather than from extensive hyperparameter tuning.

LoRA Training. We employ LoRA [S7] for training our 7B model. This decision aligns with the training strategies of baseline methods such as LISA [S10] and Text4Seg [S11], ensuring a direct and fair comparison. In contrast, our 2B model is fully fine-tuned, as its smaller size renders the efficiency benefits of LoRA [S7] less significant.

S1.2. Training Dataset

For the training data in each sub-experiment, we strictly adhere to the settings of our baselines, Text4Seg [S11] and LISA [S10], conducting experiments on the same minimal datasets (detailed in Tab. S2). This approach is intentional: we aim to demonstrate that our performance gains stem from our paradigm innovation, rather than from simply using more data. Consequently, our current training is focused on single-object grounding and segmentation tasks. While many contemporary works train on larger, mixed datasets, we are confident that *STAMP* can be scaled to these data sources in the future, presenting a clear path to even greater performance leads.

S1.3. Instruction Construction

The instruction used for training *STAMP* is included in Prompt S1 and S2.

```
Question:
o "Please segment only the [class_name] in the image.",
o "Can you segment the [class_name] in the image?",
o "Where is the [class_name] in this picture? Please respond with segmentation mask.",
o "Where is [class_name] in this image? Please output segmentation mask.",
o "Could you provide the segmentation mask for [class_name] in this image?",
o "Please segment the image and highlight [class_name]."

Answer:
* "Sure, here is the segmentation mask for [class_name] <|seg|>:",
* "Here is the segmentation mask focusing on the [class_name] <|seg|>:",
* "Here is the segmentation mask highlighting the [class_name] <|seg|>:",
* "The segmentation map for [class_name] <|seg|> is:",
* "The segmentation mask for [class_name] <|seg|> is shown below:",
* "Sure, Here's the segmentation mask for [class_name] <|seg|>:",
* "Sure, the segmented output for [class_name] <|seg|> is:",
* "Certainly, the segmentation map for [class_name] <|seg|> is:",
* "Certainly, here is the segmentation mask for [class_name]:",
* "The segmentation mask for [class_name] <|seg|> is shown below:"
```

Prompt S1. Instructions for RES task.

```
% --- 1. Segmentation Task ---
Question:
    o "Can you segment the [class_name] in this image?"
    o "Please segment the [class_name] in this image."
    o "What is [class_name] in this image ? Please output segmentation mask ."
    o "{sent} Please respond with segmentation mask."

Answer:
    * "It is <|seg|>."
    * "Sure, <|seg|>."
    * "Sure, the segmentation result is <|seg|>."
    * "<|seg|>."

% --- 2. Explanation Task ---
Question:
o "Explain [class_name] in this image."
o "Could you describe the [class_name] in the image?"
o "Tell me more about the [class_name]."

Answer:
* "[reasoning]"
```

Prompt S2. Instructions for Reasoning Segmentation task.

S2. Additional Experiments.

In this section, we list the specific design and detailed test results for evaluating *STAMP*'s efficiency.

S2.1. Implementation Setting.

For the inference latency discussed in §4.5, all methods were rigorously evaluated on a single A800 80GB GPU. Crucially, tests were conducted in isolation, ensuring only one program ran on the card during each measurement. Furthermore, all methods employed the identical test case, which is illustrated in Fig. S1.

S2.2. Results.

As shown in Tab. S3, *STAMP* excels in both accuracy and speed. *STAMP*-7B achieves the highest cIoU of 80.7, significantly outperforming SegAgent-7B(75.4) [S28] and READ-7B(71.9) [S19]. Despite high input resolution (896×896), *STAMP* remains remarkably fast: *STAMP*-2B variants (0.9s–1.3s) are comparable to embedding-based methods (e.g., LISA, PixelLM [S22]) and far faster than next-token approaches like SegAgent (15.9s). Ablation studies further confirm this efficiency; even without the mask decoder, *STAMP*-2B maintains competitive performance (75.3 cIoU) with minimal VRAM usage (10.8 GB), demonstrating superior resource efficiency.

Prompt: Please segment the trees in the image.

Figure S1. **Inference time comparison example**. This example illustrates a representative test case used to measure the inference latency across all compared methods. The consistent input allows for a fair comparison of the computational efficiency between the models discussed in Section 4.5.

S3. More Qualitative Showcases.

To comprehensively validate the capabilities of our *STAMP* model, we designed a suite of test cases targeting its performance across multiple areas, including reasoning segmentation, VQA (Visual Question Answering), multi-round conversation, multi-round segmentation, and dialogue-segmentation. For all subsequent cases, the *STAMP* model used is the *STAMP*-2B variant. This model is configured with the SAM decoder and processes inputs at an 896×896 resolution.

S3.1. Reasoning Segmentation

Reasoning Segmentation requires transcending simple recognition to integrate functional and abstract reasoning. As shown in Fig. S2, *STAMP* excels in this task by explicitly decoupling reasoning from segmentation. It first articulates inferred properties (e.g., nutritional role) before executing precise segmentation. This two-step process confirms *STAMP*'s ability to ground high-level semantic understanding directly onto pixel space, offering a distinct advantage over models that treat reasoning merely as a textual side effect.

S3.2. Visual Question Answering

Crucially, *STAMP* integrates vision-language capabilities without sacrificing the core language proficiency of the underlying MLLM. This contrasts with methods like

Table S1. **Comprehensive hyperparameter settings for all SFT stages.** We detail the training configurations for our main experiments: Referring Expression Segmentation (§4.2), Reasoning Segmentation (§4.3), and Visual Understanding (§4.4). RefCOCO* means RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17] and RefCLEF [S8].

Category	Hyperparameter	Referring Seg.		Reasoning Seg.	Dialogue & Seg.
		Qwen2-2B	Qwen2-7B	Qwen2-2B	Qwen2-2B
Training Details	Section §	§4.2	§4.2	§4.3	§4.4
	Datasets	RefCOCO*, gRefCOCO [S14] (stage 2)	RefCOCO*, gRefCOCO [S14] (stage 2)	RefCOCO*, ReasonSeg [S10]	RefCOCO*, LLaVA-665k [S15]
	Epochs	3 + 2	3 + 2	1	2
	Batch Size	96	96	96	96
	ZeRO Stage	2	2	2	2
Optimizer (AdamW)	Learning Rate	3×10^{-5}	2×10^{-5}	3×10^{-5}	3×10^{-5}
	Weight Decay	0	0	0	0
	Betas (β_1, β_2)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)	(0.9, 0.999)
	Grad Norm Clip	1.0	1.0	1.0	1.0
	Scheduler	Linear	Linear	Linear	Linear
	Warmup Ratio	0.03	0.03	0.03	0.03
LoRA Configuration	Method	Full fine-tuning (LoRA not applied)	LoRA	Full fine-tuning (LoRA not applied)	Full fine-tuning (LoRA not applied)
	Rank		64		
	Alpha (α)		128		
	Dropout		0.05		
	Target Modules		All		
Model Parameters	Total Parameters	2B	7B	2B	2B

LISA that often compromise language performance for segmentation. As shown in Fig. S3, *STAMP* remains robust across diverse VQA tasks, ranging from abstract reasoning to fine-grained grounding. This demonstrates that *STAMP* retains superior language mastery, serving as a versatile multimodal assistant beyond its primary segmentation role.

S3.3. Muti-round Dialogue

The multi-round dialogue scenario rigorously tests *STAMP*'s contextual retention and linguistic integrity. As illustrated in Fig. S4, the model's seamless performance across a six-round exchange confirms its capability as a unified conversational agent. Specifically, *STAMP* demonstrates zero-shot immunity to common MLLM failures: it successfully grounds complex temporary aliases and maintains resilience against conversational distractions. This confirms that *STAMP* preserves the critical memory and reasoning capabilities essential for conversational AI, despite the integration of segmentation.

S3.4. Muti-round Segmentation

The Multi-round Dialogue scenario evaluates *STAMP*'s contextual retention and linguistic integrity. As shown in Fig. S4, its seamless performance across a six-round exchange confirms its capability as a unified conversational agent. Specifically, *STAMP* demonstrates zero-shot robustness against common failures, successfully grounding temporary aliases and resisting conversational distractions. This sustained coherence validates that integrating segmentation preserves the critical memory and reasoning foundations required for advanced conversational AI.

S3.5. Dialogue Segmentation Interaction

The Dialogue Segmentation Interaction scenario validates *STAMP* as a unified multimodal assistant. As shown in Fig. S6, the model seamlessly transitions between linguistic responses and pixel-level masks within the same flow. A key highlight is the memory-based segmentation in Round 6, where *STAMP* grounds an object (the milk bottle) based solely on context established in Rounds 4-5. This confirms *STAMP* utilizes dialogue history as a dynamic context for visual tasks, distinguishing it from rigid, turn-by-turn frameworks.

S4. Code and Documentation.

We provide a detailed *README.md* as the primary reference for the codebase, covering essential aspects including environment setup, detailed usage protocols for training and evaluation, and an explanation of the modular project structure. Due to upload size constraints, raw image datasets are not included in this submission; however, the documentation strictly defines the expected data hierarchy. To facilitate reproducibility, we also include specific scripts designed to generate all required annotations from standard data sources.

Table S2. **Training data comparison across CPT and SFT stages.** This table contrasts the datasets used during the Continued Pre-Training (CPT) stage and the Supervised Fine-Tuning (SFT) stage for Referring Expression Segmentation. † Some methods incorporate RefCLEF during CPT, while others (like Ours, Text4Seg, *STAMP*) use it in the SFT stage.

Method	**Continued Pre-Training (CPT) Datasets**	**Referring Seg. SFT Datasets**
LISA [S10]	ADE20K [S27], COCO-Stuff [S2], PACO-LVIS [S21], PartImageNet [S6], PASCAL-Part [S3], RefCLEF† [S8], RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], LLaVA-665k [S15]	RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17]
PixelLM [S22]	ADE20K [S27], COCO-Stuff [S2], PACO-LVIS [S21], RefCLEF† [S8], RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], LLAVA-150k [S15], MUSE [S22]	**None**
GSVA [S26]	ADE20K [S27], COCO-Stuff [S2], PACO-LVIS [S21], Mapillary Vistas [S18], PASCAL-Part [S3], RefCLEF† [S8], RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], gRefCOCO [S14], LLaVA-Instruct-150K [S15], ReasonSeg [S10]	RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17]
READ [S19]	ADE20K [S27], COCO-Stuff [S2], PACO-LVIS [S21], PASCAL-Part [S3], RefCLEF† [S8], RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], LLaVA-Instruct-150K [S15]	R-RefCOCO [S24], FP-RefCOCO+ [S25], FP-RefCOCOg [S25]
Seg-Zero [S16]	**None**	RefCOCOg [S25]
SegLLM [S23]	RefCOCO [S8], Visual Genome [S9], PACO-LVIS [S21], LVIS [S5], Pascal Panoptic Part [S4], ADE20K [S27], COCO-Stuff [S2], Attributes-COCO [S13], ReasonSeg [S10], MRSeg (hard) [S23]	**None**
SegAgent [S28]	**None**	DIS5K [S20], ThinObject5K [S12]
Text4Seg [S11]	**None**	RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], RefCLEF† [S8]
STAMP	**None**	RefCOCO [S8], RefCOCO+ [S8], RefCOCOg [S17], RefCLEF† [S8]

Table S3. **Ablation and efficiency study on the single-object RES task.** We compare performance on the RES task (cIoU), inference time, and decoder dependency across paradigms. Methods are grouped by their core segmentation paradigm as defined in Table 1. All inference metrics (Time and VRAM) are benchmarked on an A800 GPU.

Method	**Input Size**	**cIoU**	**Time (s)**	**Decoder Status**	**Infer VRAM (GB)**
Paradigm 1: Embedding Prediction					
LISA-7B (CVPR'24) [S10]	336 × 336	69.9	1.1	Required SAM	15.5
LISA-13B (CVPR'24) [S10]	336 × 336	70.1	1.8	Required SAM	27.6
READ-7B (CVPR'25) [S19]	336 × 336	71.9	1.5	Required SAM	15.7
PixelLM-7B (CVPR'24) [S22]	448 × 448	69.2	0.7	Required Custom Dec.	14.1
Paradigm 2: Next-token Prediction					
Seg-Zero-7B (arXiv'25) [S16]	840 × 840	74.1	4.5	Required SAM2	18.0
SegAgent-7B (CVPR'25) [S28]	448 × 448	75.4	15.9	Required SAM	20.7
Text4Seg-7B (ICLR'25) [S11]	336 × 336	74.9	4.0	Optional	20.4
W/O Mask Decoder	336 × 336	70.9	3.9	*SAM*	20.4
Paradigm 3: All-mask Prediction (Ours)					
STAMP-2B	504 × 504	77.1	0.9	Optional	10.8
STAMP-2B	726 × 726	78.6	1.1	Optional	10.8
STAMP-2B	896 × 896	79.1	1.3	Optional	10.8
W/O Mask Decoder	896 × 896	75.3	0.9	*SAM*	10.8
STAMP-7B	896 × 896	80.7	2.4	Optional	25.6
W/O Mask Decoder	896 × 896	76.2	2.1	*SAM*	25.6

User

In our daily lives, we need to consume ***food to replenish vitamins, dietary fiber, water, etc***. What objects in the picture, when ingested, can do this?

STAMP *Please explain.*

The object in the picture that can do this is ***a bowl of fruit***, such as the one containing bananas, limes, and other fruits. ***Consuming a variety of fruits provides essential nutrients and vitamins that are vital for maintaining good health and overall well-being***. The bowl of fruit in the image serves as a convenient and visually appealing way to incorporate these nutrients into one's diet.

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

User

Hey, in this kitchen photo, there are a bunch of different items. Could you tell me which object is the one that is used for ***telling time*** and its location? Please segment it.

STAMP *Please explain.*

The object used for ***telling time in this kitchen is the clock***, and it is located on the wall.

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

User

Somthing that can be used to eat things. Please segment it.

STAMP *Please explain.*

The image shows a fork resting on a glass bowl filled with a mixture of ingredients. The ***fork is used to scoop the mixture***, which appears to be a mixture of fish and vegetables. The bowl is placed on a surface, and the fork is positioned on top of the mixture, ready to be used to eat the contents. ***The fork is likely designed for serving and eating purposes, as it is designed to hold and manipulate food items.***

STAMP *Please segment it.*

Sure, the segmentation result is<|seg|>.

Figure S2. **Reasoning segmentation cases.** The left column presents the model's detailed explanations of inferred object properties, such as the nutritional role of fruit, the time-keeping function of a clock, and the utility of a fork. Correspondingly, the right column displays the successful segmentation results derived from this understanding."

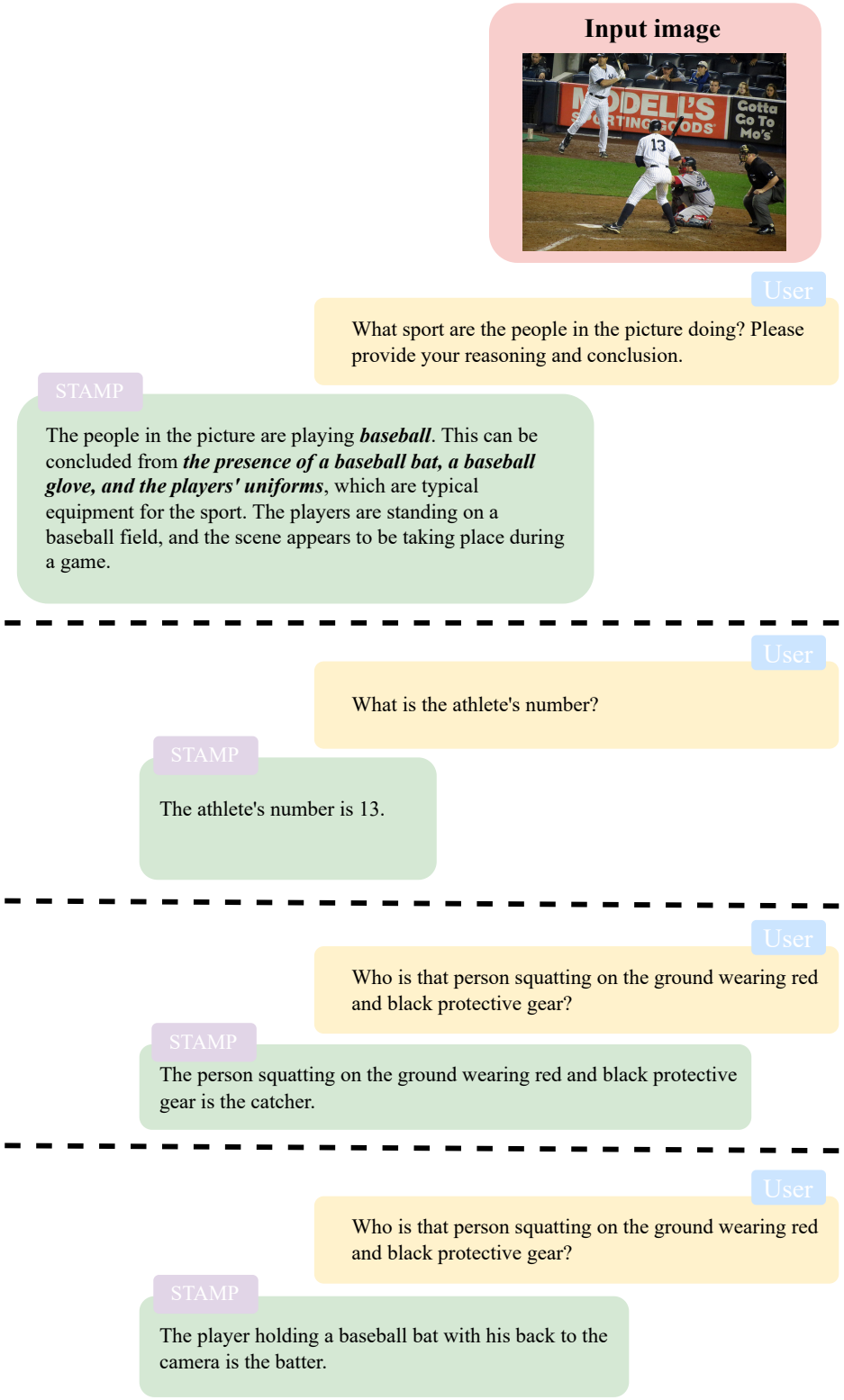

Figure S3. **Visual question answering (VQA) cases.** This example showcases *STAMP*'s performance across various single-turn VQA scenarios. Each exchange, separated by a dashed line, is treated as an independent question-answering task. The model demonstrates comprehensive visual understanding, covering: 1) complex scene reasoning (identifying the sport), 2) precise detail extraction (athlete's number), and 3) specific referential grounding (identifying the catcher and batter).

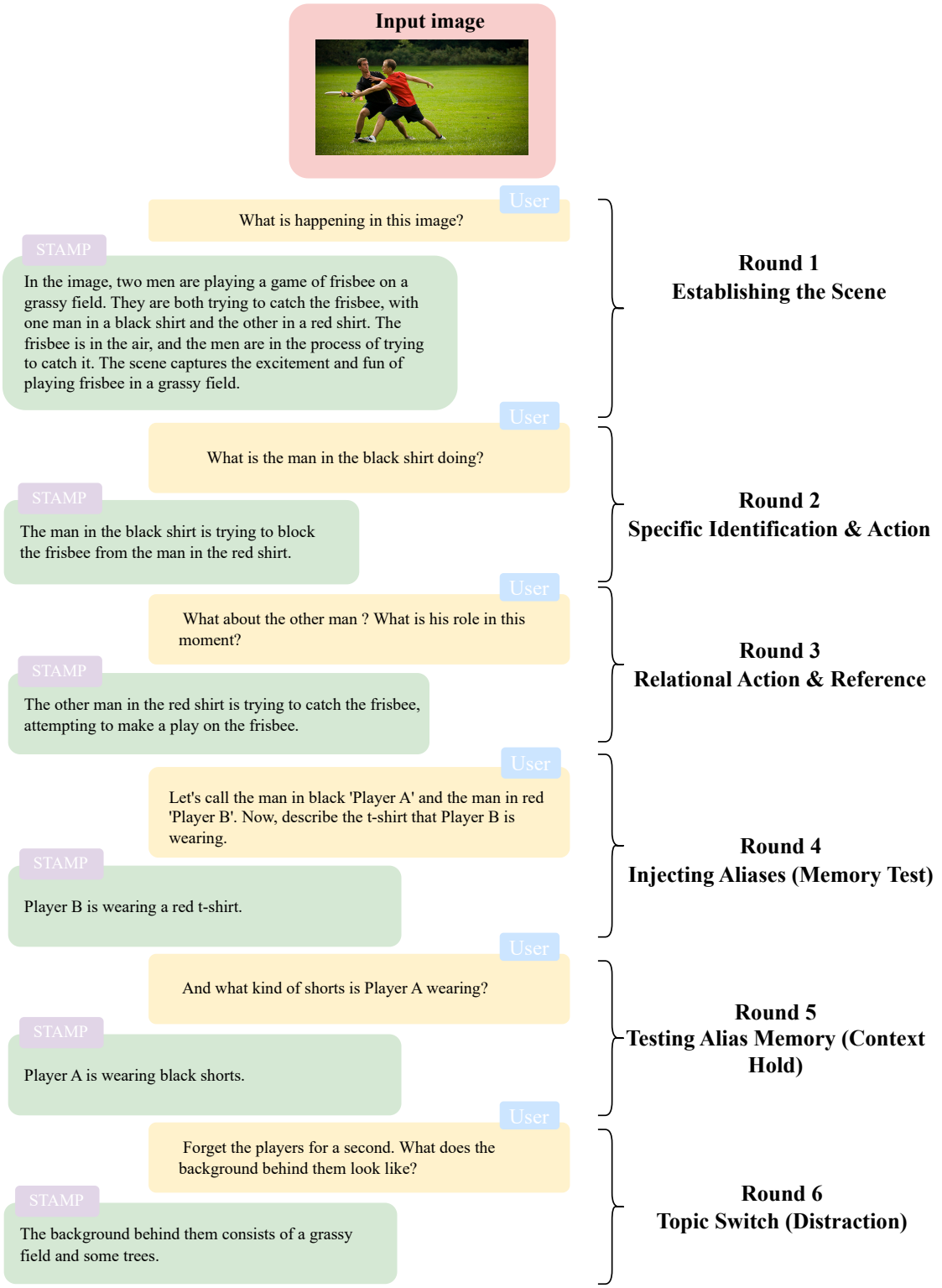

Figure S4. **Multi-round conversational capabilities.** This figure illustrates *STAMP*'s performance in a complex, multi-turn conversation about a single image. The interaction is structured to test the model's ability to maintain context, including: establishing the scene (Round 1), specific identification (Round 2), relational referencing (Round 3), memory injection/testing via aliases (Rounds 4-5), and topic switching with distraction (Round 6). Successful completion of these rounds validates *STAMP*'s robust contextual memory and dialogue coherence.

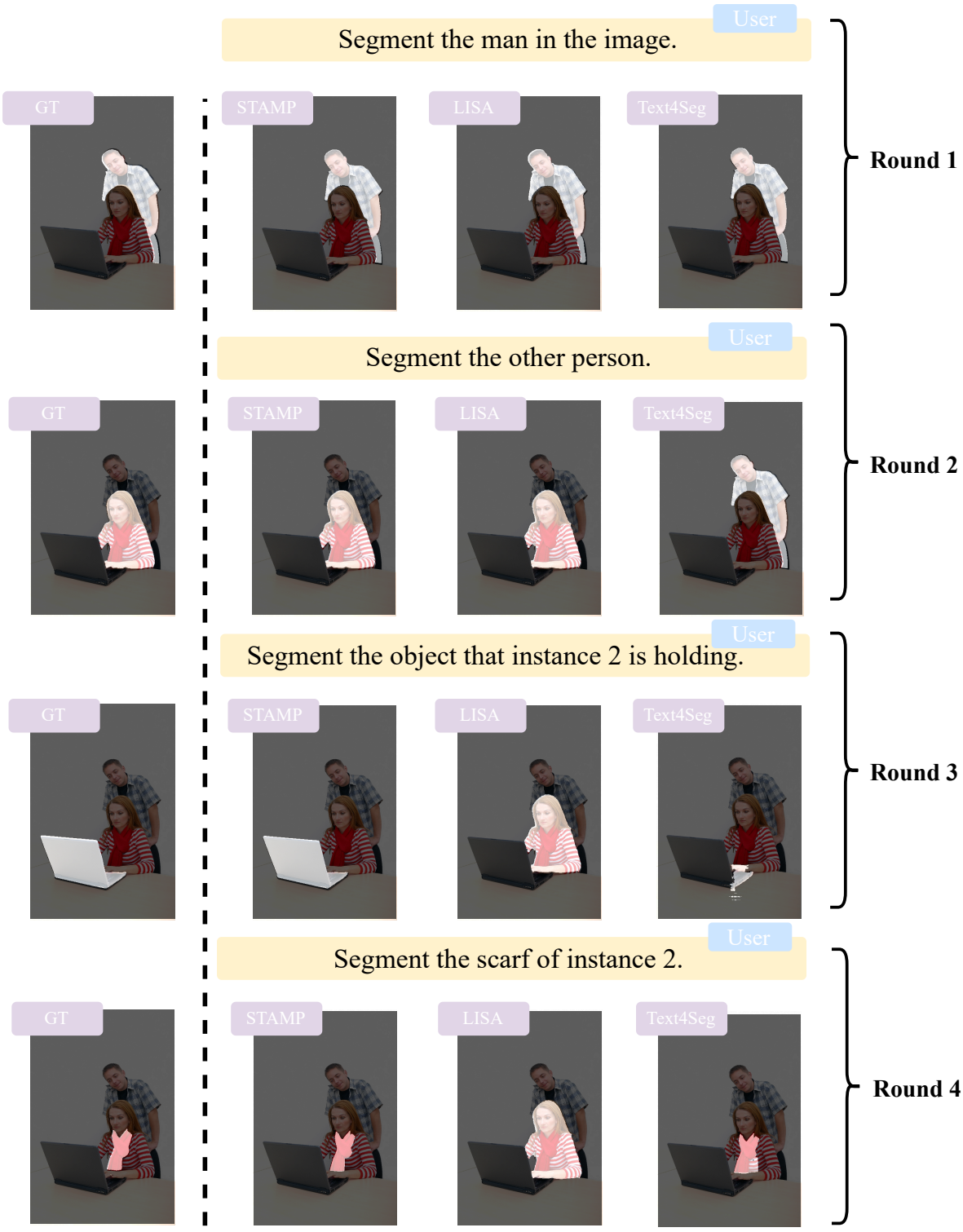

Figure S5. **Multi-round segmentation cases.** This figure illustrates *STAMP*'s performance in multi-turn interactive segmentation, compared against LISA and Text4Seg. The sequential rounds (Round 1 to Round 4) require the model to maintain segmentation context and handle complex referential queries. *STAMP* consistently delivers accurate segmentation even as the queries become more nuanced (e.g., referencing "the other person" in Round 2, segmenting an object held by a previously identified instance in Round 3, and fine-grained attribute segmentation like "scarf of instance 2" in Round 4), showcasing its superior contextual understanding compared to other methods.

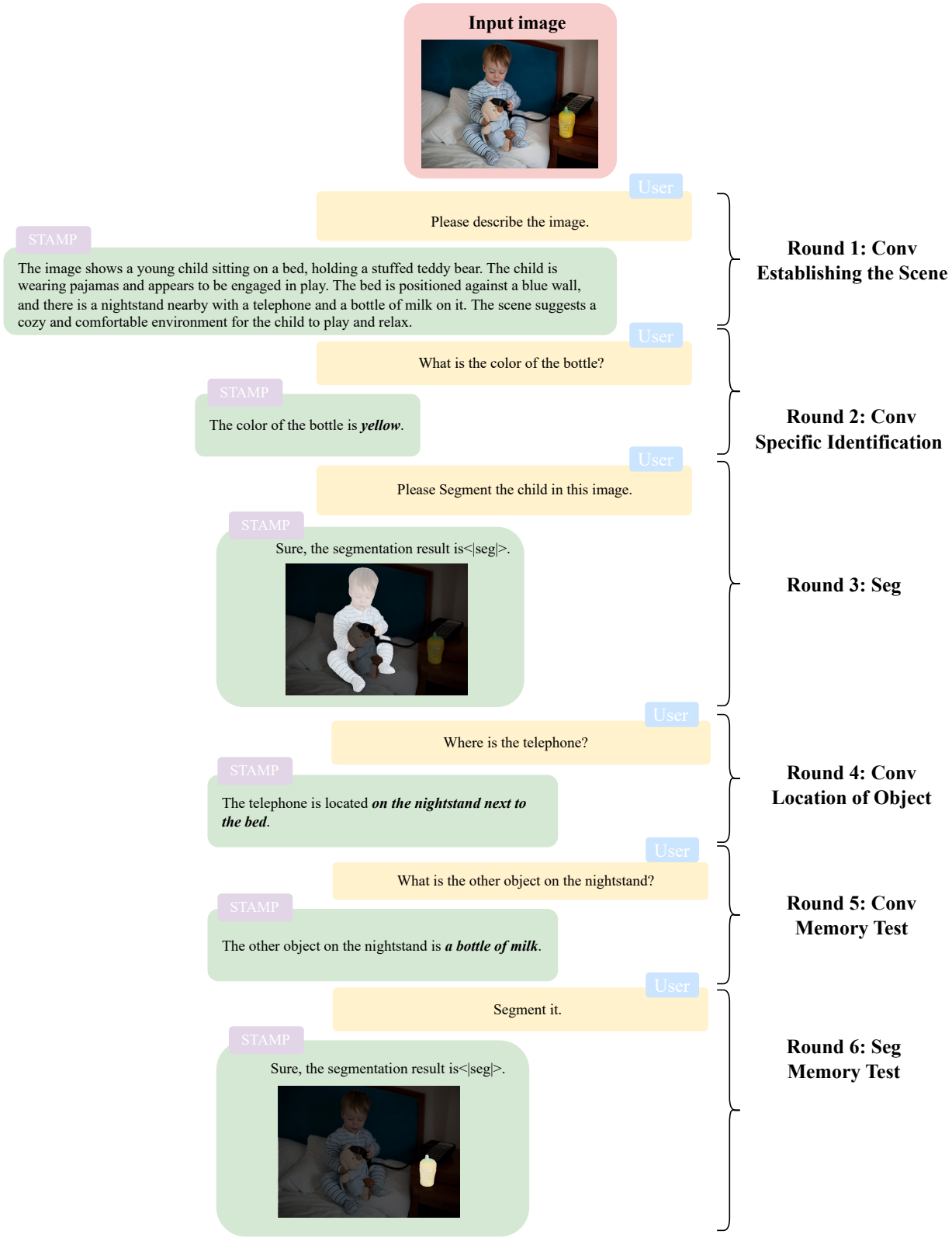

Figure S6. **Seamless interleaving of segmentation and dialogue.** It shows *STAMP*'s unique ability to flexibly interleave Conv and Seg tasks in one dialogue thread. This structure demonstrates context-dependent segmentation, culminating in memory-based segmentation (Round 6). The successful segmentation in Round 6 confirms *STAMP* accurately grounds an object (bottle of milk) into pixel space using only the dialogue history (Rounds 4-5), without requiring a new descriptive prompt.

References

[S1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[S2] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. COCO-Stuff: Thing and stuff classes in context. In *CVPR*, 2018. 4

[S3] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1971–1978, 2014. 4

[S4] Daan De Geus, Panagiotis Meletis, Chenyang Lu, Xiaoxiao Wen, and Gijs Dubbelman. Part-aware panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5485–5494, 2021. 4

[S5] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019. 4

[S6] Ju He, Shuo Yang, Shaokang Yang, Adam Kortylewski, Xiaoding Yuan, Jie-Neng Chen, Shuai Liu, Cheng Yang, Qihang Yu, and Alan Yuille. Partimagenet: A large, high-quality dataset of parts. In *European Conference on Computer Vision*, pages 128–145. Springer, 2022. 4

[S7] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 1

[S8] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *EMNLP*, 2014. 3, 4

[S9] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 4

[S10] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: Reasoning segmentation via large language model. In *CVPR*, 2024. 1, 3, 4

[S11] Mengcheng Lan, Chaofeng Chen, Yue Zhou, Jiaxing Xu, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Text4Seg: Reimagining image segmentation as text generation. In *ICLR*, 2025. 1, 4

[S12] Jun Hao Liew, Scott Cohen, Brian Price, Long Mai, and Jiashi Feng. Deep interactive thin object selection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 305–314, 2021. 4

[S13] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[S14] Chang Liu, Henghui Ding, and Xudong Jiang. GRES: Generalized referring expression segmentation. In *CVPR*, 2023. 3, 4

[S15] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 3, 4

[S16] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-Zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. 4

[S17] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *CVPR*, 2016. 3, 4

[S18] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE international conference on computer vision*, pages 4990–4999, 2017. 4

[S19] Rui Qian, Xin Yin, and Dejing Dou. Reasoning to attend: Try to understand how <SEG> token works. In *CVPR*, 2025. 2, 4

[S20] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *European Conference on Computer Vision*, pages 38–56. Springer, 2022. 4

[S21] 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 *CVPR*, 2023. 4

[S22] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. PixelLM: Pixel reasoning with large multimodal model. In *CVPR*, 2024. 2, 4

[S23] XuDong Wang, Shaolun Zhang, Shufan Li, Kehan Li, Konstantinos Kallidromitis, Yusuke Kato, Kazuki Kozuka, and Trevor Darrell. SegLLM: Multi-round reasoning segmentation with large language models. In *ICLR*, 2025. 4

[S24] Jianzong Wu, Xiangtai Li, Xia Li, Henghui Ding, Yunhai Tong, and Dacheng Tao. Toward robust referring image segmentation. *IEEE Transactions on Image Processing*, 33:1782–1794, 2024. 4

[S25] 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. In *CVPR*, 2024. 4

[S26] Zhuofan Xia, Dongchen Han, Yizeng Han, Xuran Pan, Shiji Song, and Gao Huang. GSVA: Generalized segmentation via multimodal large language models. In *CVPR*, 2024. 4

[S27] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 4

[S28] Muzhi Zhu, Yuzhuo Tian, Hao Chen, Chunluan Zhou, Qingpei Guo, Yang Liu, Ming Yang, and Chunhua Shen.

SegAgent: Exploring pixel understanding capabilities in mllms by imitating human annotator trajectories. In *CVPR*, 2025. 2, 4