Conversational Image Segmentation: Grounding Abstract Concepts with Scalable Supervision

Supplementary Material

Contents

A Qualitative Results 1
- A.1 Qualitative Predictions by ConverSeg-Net 1
- A.2 Failure Cases 1
- A.3 Annotation Quality in RefCOCO/+/g 1

B Conversational Data Engine Details 1
- B.1 Meta-Prompts and Stage Details 1

C Benchmark Construction and Analysis 2
- C.1 Annotation Protocols 2
- C.2 VLM Verifier Reliability for ConverSeg . . . 3
- C.3 Additional Statistics and Visualizations 3
- C.4 Additional Qualitative Examples 5

D Implementation Details 5
- D.1 Architecture 5
- D.2 Training Hyperparameters 5

E Additional Quantitative Results 5
- E.1 Cumulative IoU (cIoU) 5
- E.2 Additional Baselines 8

A. Qualitative Results

In this section, we provide additional qualitative examples of predictions by ConverSeg-Net.

A.1. Qualitative Predictions by ConverSeg-Net

We compare ConverSeg-Net with LISA [4] instantiated with LLaVA-7B and Llama2-13B backbones. Note that ConverSeg-Net uses a Qwen2.5-VL 3B backbone, which is considerably smaller than both LISA variants.

In Figs. 13 to 15, we show model predictions on images from the human-annotated split of ConverSeg. In Figs. 16 to 18, we show predictions on the SAM-seeded split of ConverSeg. Across both splits, ConverSeg-Net typically produces masks that more closely match the conversational intent of the prompt, accurately identifying and segmenting the requested regions despite its smaller backbone size.

We also explore out-of-distribution (OOD) behavior in Fig. 1, where we show qualitative predictions on images from the DROID dataset [3] and the Warehouse dataset [5]. In both settings, ConverSeg-Net often localizes the regions implied by the prompts, suggesting prospective applications in domestic and warehouse robotics.

A.2. Failure Cases

In Figs. 19 and 20, we present representative failure cases of ConverSeg-Net on the human-annotated and SAM-seeded splits of ConverSeg, respectively. We observe several recurring failure modes. For ambiguous prompts such as "Segment the object reflected by the window glass" in Fig. 20, ConverSeg-Net segments the reflection of the person rather than the person itself. In other cases, the model selects only one of several valid targets, yielding high precision but low recall; examples include "Identify cylindrical vessels designed for dry ingredient storage" in Fig. 19 and "Segment signs pointing diagonally upward" in Fig. 20. Additional diverse failure cases illustrating similar behaviors are shown in the figures below.

A.3. Annotation Quality in RefCOCO/+/g

RefCOCO/+/g datasets contain noisy ground truth annotations with incorrect or inaccurate masks, which can artificially lower performance metrics. Fig. 2 shows representative examples where ConverSeg-Net produces semantically reasonable predictions that receive low gIoU scores due to problematic ground truth annotations. Common issues include ground truth masks including irrelevant regions, or have poor boundary alignment despite correct semantic interpretation. In such cases, the gIoU metric penalizes reasonable model predictions, leading to artificially deflated performance numbers. These examples illustrate that numerical results on RefCOCO/+/g should be interpreted with caution, as the annotation quality does not always reflect the true segmentation difficulty or model capability.

B. Conversational Data Engine Details

In this section, we expand on the five-stage conversational data engine introduced in Section 5.1 of the main paper. Recall that the engine automatically constructs conversational segmentation triplets (image, prompt, mask) from COCO images, and is used to generate the 61K training examples for ConverSeg-Net. Below we provide the meta-prompts and additional implementation details for each stage.

B.1. Meta-Prompts and Stage Details

Stage 1: Scene Understanding. In Stage 1 we obtain rich region-level descriptions that serve as the semantic backbone for subsequent stages. Figure 8 shows the meta-prompt used to query Gemini-2.5-Flash for these region descriptions, given the input image.

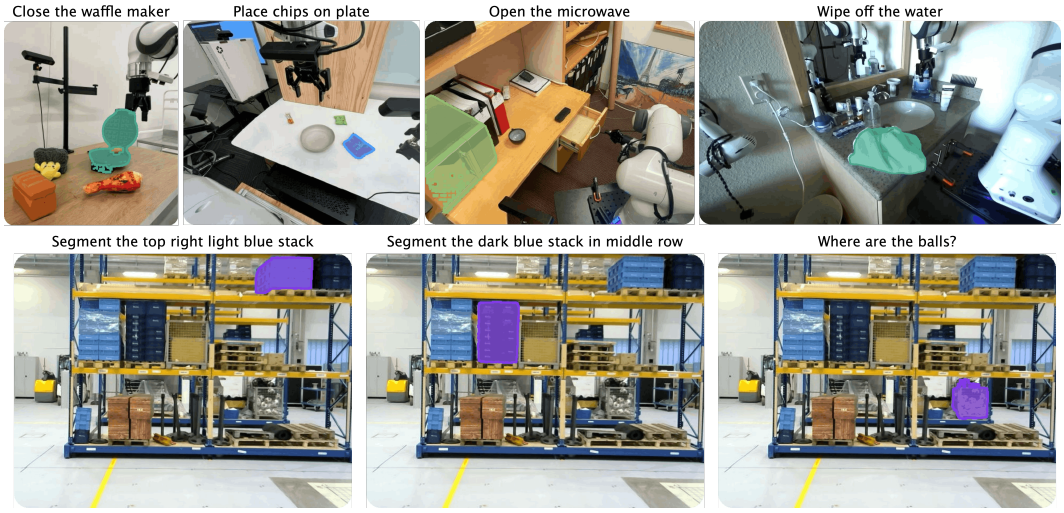


Figure 1. **Out-of-distribution qualitative examples.** Predictions of CONVERSEG-NET on images from the DROID dataset [3] and the Warehouse dataset [5]. For each example, we show the input image, conversational prompt, and the predicted mask overlaid. CONVERSEG-NET often localizes the regions implied by the prompts despite the distribution shift, hinting at prospective applications in domestic and warehouse robotics.

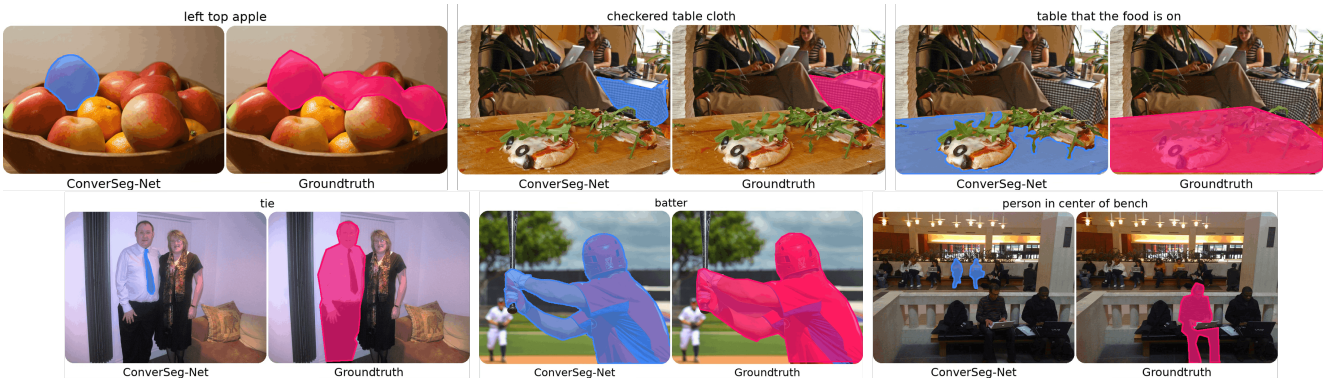


Figure 2. **Noisy annotations in RefCOCO/+/g.** CONVERSEG-NET predictions (blue) are semantically reasonable but receive low gIoU due to problematic ground truth masks (pink): incomplete coverage, irrelevant regions, or poor boundaries.

Stage 2: Mask Generation. In Stage 2 we convert textual region descriptions into segmentation masks. We query the Moondream model with its default API configuration to predict bounding boxes, and then pass these boxes to SAM2 to obtain corresponding masks. Since this stage does not rely on any additional natural-language control, we do not use any dedicated meta-prompt here.

Stage 3: Mask Quality Verification. Stage 3 filters and refines the SAM2 masks using VLM-based checks. Figure 9 shows the meta-prompt used for the *mask–text consistency* check, where the VLM judges whether a candidate mask matches the associated region description. Figure 10 shows the meta-prompt used for *mask refinement and selection*, where the VLM compares two candidate masks and selects the best one.

Stage 4: Concept-Driven Prompt Generation. Stage 4 converts region descriptions into conversational prompts anchored in our five concept families. We use a separate concept-specific meta-prompt for each family. Fig. 11 shows the meta-prompt for the **affordances & functions** concept. The meta-prompts for the remaining concepts follow the same structure; we omit them here to avoid redundancy.

Stage 5: Prompt–Mask Alignment Verification. Finally, Stage 5 verifies that the generated conversational prompt is aligned with the selected mask. Fig. 12 shows the meta-prompt used for this verification step, where the VLM judges whether the prompt correctly and unambiguously describes the masked region.

C. Benchmark Construction and Analysis

C.1. Annotation Protocols

We describe the interface and instructions given to human annotators for constructing CONVERSEG. As discussed in Section 4 of the main paper, CONVERSEG is obtained via *human verification* of examples produced by the conversa-

tional data engine. For each candidate example, annotators were shown the input image with the AI-generated mask overlaid and the corresponding conversational prompt. The original image without the mask overlaid was also provided for context.

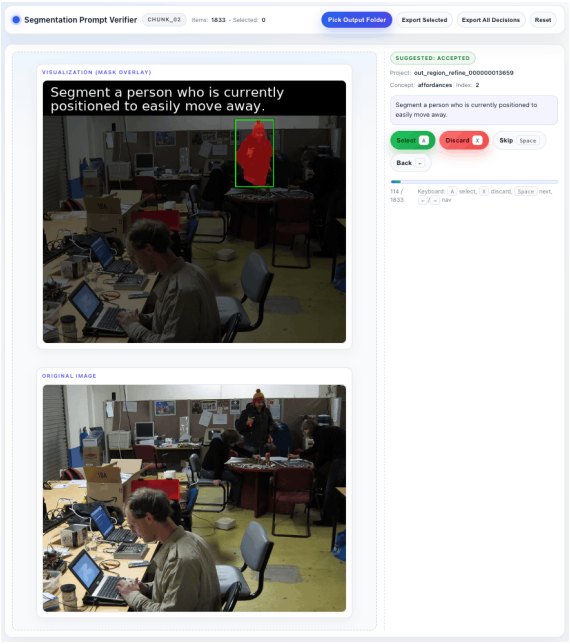


Figure 3. **Annotation interface for constructing CONVERSEG.** Annotators are shown the input image with the AI-generated mask overlaid, along with the corresponding conversational prompt and the suggested decision from the AI verifier. They then decide whether the prompt and mask are semantically aligned and select either *Accept/Select* or *Reject/Discard*; rejected examples are discarded without further editing.

The user interface was intentionally kept simple. Annotators were asked to judge whether the prompt and mask were semantically aligned, i.e., whether the mask accurately and sufficiently covered all regions referred to by the prompt without including substantial irrelevant areas. They then chose between two options: *Accept* (if the example was valid) or *Reject* (otherwise). The decision proposed by the AI verifier was also displayed as a suggested label, but annotators were free to override it. Rejected examples were simply discarded; we did not ask annotators to refine or edit masks. A screenshot of the annotation interface is shown in Fig. 3.

C.2. VLM Verifier Reliability for CONVERSEG

To better understand how the VLM verifier behaves, Fig. 4 shows qualitative examples from three categories: (1) the VLM *accepts* but the human *rejects*; (2) both the VLM and the human *reject*; (3) the VLM *rejects* but the human *accepts*.

In the first case, the VLM typically accepts examples where the mask *partially* satisfies the prompt. For instance, for the prompt "Identify the luggage sturdy enough to use as a step", the VLM accepts a mask highlighting two pieces of luggage, even though additional items could also reasonably satisfy the prompt. In other cases, disagreement is driven by mask quality rather than semantics; for example, for "Segment the objects currently providing thermal insulation", the mask includes the blanket but also the person, leading the human to reject the example despite the VLM accepting it.

In the second case, where both the VLM and the human reject an example, the VLM reliably identifies clear errors (e.g., severe under-/over-coverage or obvious semantic mismatch), and its decision closely matches the human judgment.

In the third case, where the VLM rejects but the human accepts, the prompts are often somewhat ambiguous. For example, for "Segment the object reflected by the window glass", the VLM expects the reflection itself to be masked and therefore rejects the example, while the human accepts a mask covering the physical object.

For benchmark construction, and to avoid any single instance dominating the dataset with many similar prompts, annotators were also instructed to reject prompts referring to duplicate objects, even if the prompt–mask pair was otherwise accurate. These rejected pairs remain useful for training but are excluded from CONVERSEG to preserve diversity.

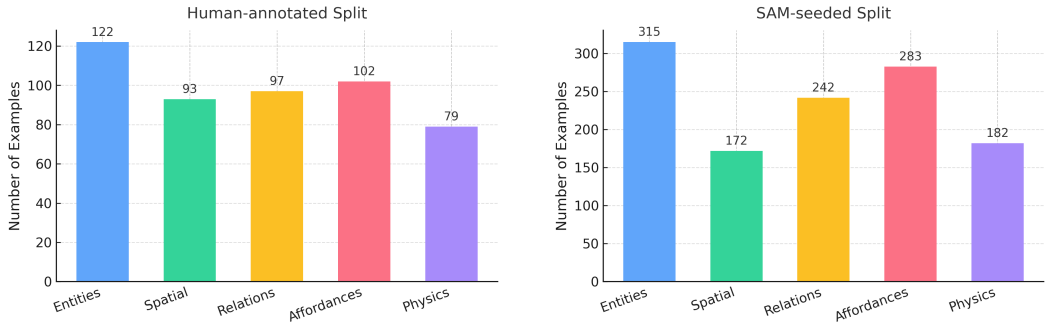
Aggregating across these conditions, the VLM verifier and human annotators make the same decision on about 70% of examples. In the common disagreement case where the VLM *accepts* but the human *rejects*, the VLM decision is often not semantically incorrect (e.g., partial coverage or duplicate prompts), so these examples remain valuable as training data. This behavior is appropriate for automatically generating a large pool of candidate examples, while human verification is used to ensure benchmark-quality data. In practice, the verifier provides a strong starting set from which annotators can efficiently curate high-quality examples for CONVERSEG.

C.3. Additional Statistics and Visualizations

In Fig. 5, we show bar charts indicating the number of examples per concept family (entities, spatial & layout, relations & events, affordances & functions, physics & safety) for each split of CONVERSEG. These statistics complement the distributional analysis in the main paper and confirm that all concept families are well represented.

Region Type Diversity. Beyond whole-object instances, CONVERSEG includes diverse region types such as object parts, surfaces, and functional areas. The SAM-seeded split naturally incorporates these non-instance regions because SAM2 can generate masks for parts and surfaces in addition to complete objects. The human-annotated split further incorporates "stuff" regions from COCO-Panoptic, such as sky,

Figure 4. **Qualitative examples of VLM verifier behavior,** illustrating agreement and disagreement with human annotators on candidate prompt–mask pairs in ConverSeg.



(a) Human-annotated split

(b) SAM-seeded split

Figure 5. Distribution of examples per concept in the two splits of CONVERSEG.

Figure 6. Examples of non-instance regions in CONVERSEG. The masks capture object parts, surfaces, and functional areas, demonstrating coverage of diverse region types beyond complete object instances.

grass, walls, and other amorphous areas. Fig. 6 shows representative examples from the SAM-seeded split, including object parts, surfaces, and functional regions.

C.4. Additional Qualitative Examples

We provide additional qualitative examples from CONVERSEG in Fig. 7, illustrating the diversity of conversational prompts and corresponding masks across concept families and splits.

D. Implementation Details

D.1. Architecture

Prompt encoder. We use Qwen2.5-VL-3B [1] as a frozen multimodal backbone: it takes both the RGB image and the conversational prompt as input. From the final hidden states we keep only positions corresponding to text tokens (padding, special tokens, and image placeholders are discarded). These token embeddings are layer-normalized and linearly projected to the SAM2 decoder width to form sparse language tokens. The EOS embedding is passed through a small MLP and broadcast as a $C \times H \times W$ dense bias map. Only the adapter projections and LoRA weights on top of Qwen are trained.

Mask decoder. We use the SAM2 [6] Hiera-L configuration (`sam2_hiera_l.yaml`). The image encoder is frozen. The mask decoder takes the sparse and dense language embeddings as prompt inputs. We supervise only the first output mask.

Hyperparameter	Value
Optimizer	AdamW
Learning rate (η_1, η_2)	1×10^{-4}, 1×10^{-5}
LR schedule	Warmup + cosine (min 10^{-6})
Weight decay	0.05
Batch size / grad. accum.	6 / 1
Steps per stage	35 000
Image resolution	1024×1024 (longer side)
LoRA rank and alphs (r, α)	16, 32

Table 1. Training hyperparameters for pretraining (η_1) and conversational post-training (η_2).

D.2. Training Hyperparameters

We fine-tune the SAM2 mask decoder, SAM2 prompt encoder, and language adapter in two stages (pretraining and conversational post-training). Each stage is trained for 35 000 steps with AdamW and a cosine schedule with linear warmup. Images are resized so that the longer side is 1024 pixels; masks are binarized and eroded with a 5×5 kernel (one iteration). We use a batch size of 6 with no gradient accumulation. The main optimization and model hyperparameters are summarized in Tab. 1.

E. Additional Quantitative Results

In this section, we present additional quantitative comparisons between CONVERSEG-NET and existing baselines.

E.1. Cumulative IoU (cIoU)

In the main paper, we reported gIoU performance of CONVERSEG-NET on the RefCOCO/+/g and ReasonSeg benchmarks. In Tab. 4, we report the corresponding cumulative IoU (cIoU) for CONVERSEG-NET on the same benchmarks, complementing the gIoU results. In Tab. 3, we

Figure 7. **Additional qualitative examples from ConverSeg.** Each panel shows an input image, its conversational prompt, and the corresponding ground-truth mask overlaid. Examples span all concept families (entities, spatial & layout, relations & events, affordances & functions, physics & safety) and belong to the human-annotated split.

additionally report cIoU performance of ConverSeg-Net on ConverSeg.

Model	Prompt Encoder	RefCOCO			RefCOCO+			RefCOCOg			ReasonSeg			
		val	testA	testB	val	testA	testB	val(U)	test(U)	val(G)	val	test	test(short)	test(long)
LISA	LLaVA 7B	74.9	79.1	72.3	65.1	70.8	58.1	67.9	70.6	–	44.4	36.8	37.6	36.6
LISA⋆	LLaVA 7B	–	–	–	–	–	–	–	–	–	52.9	47.3	40.6	49.4
LISA	LLaVA 13B	–	–	–	–	–	–	–	–	–	48.9	44.8	39.9	46.4
LISA⋆	LLaVA 13B	–	–	–	–	–	–	–	–	–	56.2	51.7	44.3	54.0
LISA⋆	Llama2 13B	–	–	–	–	–	–	–	–	–	60.0	51.5	43.9	54.0
LISA	LLaVA1.5 7B	–	–	–	–	–	–	–	–	–	53.6	48.8	48.3	49.2
LISA⋆	LLaVA1.5 7B	–	–	–	–	–	–	–	–	–	61.3	55.6	48.3	57.9
LISA	LLaVA1.5 13B	–	–	–	–	–	–	–	–	–	57.7	53.8	50.8	54.7
LISA⋆	LLaVA1.5 13B	–	–	–	–	–	–	–	–	–	65.0	61.3	55.4	63.2
SEEM	–	–	–	–	–	–	–	–	75.1	–	25.5	24.3	20.1	25.6
Grounded SAM	–	–	–	–	–	–	–	–	–	–	26.0	21.3	17.8	22.4
OVSeg	–	–	–	–	–	–	–	–	–	–	28.5	26.1	18.0	28.7
Seg-Zero	Qwen2.5-VL 3B	–	–	–	–	–	–	–	–	–	58.2	56.1	–	–
Seg-Zero	Qwen2.5-VL 7B	–	–	–	–	–	–	–	–	–	62.6	57.5	–	–
GSVA⋆	Vicuna 13B	79.2	81.7	77.1	70.3	73.8	63.6	75.7	77.0	–	–	–	–	–
GLaMM	Vicuna 7B	79.5	83.2	76.9	72.6	78.7	64.6	74.2	74.9	–	47.4	–	–	–
SAM4MLLM	Qwen-VL 7B	–	–	–	–	–	–	–	–	–	46.7	–	–	–
SAM4MLLM	LLaVA1.6 7B	79.6	82.8	76.1	73.5	77.8	65.8	74.5	75.6	–	–	–	–	–
SAM4MLLM	LLaVA1.6 8B	79.8	82.7	74.7	74.6	80.0	67.2	75.5	76.4	–	58.4	–	–	–
GLEE	CLIP	79.5	–	–	68.3	–	–	70.6	–	–	–	–	–	–
GLEE	CLIP	80.0	–	–	69.6	–	–	72.9	–	–	–	–	–	–
DETRIS-L	CLIP	81.0	81.9	79.0	75.2	78.6	70.2	74.6	75.3	–	–	–	–	–
UniLSeg-20	CLIP ViT-B/16	80.5	81.8	78.4	72.7	77.0	67.0	78.4	79.5	–	–	–	–	–
UniLSeg-100	CLIP ViT-B/16	81.7	83.2	79.9	73.2	78.3	68.2	79.3	80.5	–	–	–	–	–
PSALM	Phi1.5 1.3B	83.6	84.7	81.6	72.9	75.5	70.1	73.8	74.4	–	–	–	–	–
EVF-SAM†	BEIT-3-Large	82.1	83.7	80.0	75.2	78.3	70.1	76.8	77.4	–	–	–	–	–
EVF-SAM‡	BEIT-3-Large	82.4	84.2	80.2	76.5	80.0	71.9	78.2	78.3	–	–	–	–	–
RICE	Qwen2.5-7B	83.5	85.3	81.7	79.4	82.8	75.4	79.8	80.4	–	–	–	–	–
MLCD-seg	Qwen2.5-7B	83.6	85.3	81.5	79.4	82.9	75.6	79.7	80.5	–	–	–	–	–
HyperSeg	Phi2 2.7B	84.8	85.7	83.4	79.0	83.5	75.2	79.4	78.9	–	–	–	–	–
HyperSeg	Phi2 3B	–	–	–	–	–	–	–	–	–	59.2	–	–	–
Gemini Seg	Gemini2.5 Flash	–	–	–	–	–	–	–	–	–	28.3	30.6	16.5	35.0
X-SAM	Phi3 3.8B	85.1	87.1	83.4	78.0	81.0	74.4	83.8	83.9	–	56.6	57.8	47.7	56.0
RSVP	LLaVA1.6 7B	–	–	–	–	–	–	–	–	–	59.2	56.9	47.9	58.4
RSVP	Qwen2-VL 7B	–	–	–	–	–	–	–	–	–	58.6	56.1	48.5	57.1
RSVP	Gemini1.5-Flash	–	–	–	–	–	–	–	–	–	56.9	57.1	47.3	60.2
RSVP	GPT-4o	–	–	–	–	–	–	–	–	–	64.7	60.3	55.4	61.9
ConverSeg-Net (Base)	Qwen2.5-VL 3B	79.2	81.3	76.7	73.8	78.4	68.5	74.0	74.4	74.5	52.2	49.9	46.6	50.9
ConverSeg-Net	Qwen2.5-VL 3B	79.9	81.3	76.8	74.0	79.0	69.5	74.7	74.9	75.3	59.5	55.1	50.4	56.6
ConverSeg-Net	Qwen2.5-VL 7B	79.8	81.8	77.5	75.0	79.7	70.5	75.4	76.0	76.0	59.8	58.7	52.2	60.8

Table 2. **Referring expression segmentation (gIoU, %).** ConverSeg-Net is competitive on RefCOCO/+/g and achieves strong zero-shot performance on ReasonSeg, surpassing methods fine-tuned on ReasonSeg (⋆). † trained on RefCOCO only; ‡ on RefCOCO plus additional datasets (Objects365, PACO-LVIS, PASCAL-Part, etc).

Model	Prompt Encoder	SAM-seeded (cIoU)						Human-annotated (cIoU)					
		All	Ent.	Spat.	Rel.	Aff.	Phys.	All	Ent.	Spat.	Rel.	Aff.	Phys.
ConverSeg-Net	Qwen2.5-VL 3B	71.0	75.1	77.5	75.1	63.7	61.3	66.1	70.8	63.3	68.2	67.4	54.7
ConverSeg-Net	Qwen2.5-VL 7B	72.1	75.7	75.7	77.7	66.1	62.9	69.5	72.9	68.9	72.6	67.0	62.2

Table 3. **ConverSeg benchmark results (cIoU, %).** Each subset reports performance across the five concept categories – Entities, Spatial, Relations, Affordances, and Physics & Safety – and summarizes across all (*All*).

Model	Prompt Encoder	RefCOCO			RefCOCO+			RefCOCOg			ReasonSeg			
		val	testA	testB	val	testA	testB	val(U)	test(U)	val(G)	val	test	test(short)	test(long)
ConverSeg-Net	Qwen2.5-VL 3B	79.7	81.0	76.7	74.1	78.0	69.3	75.5	75.4	77.1	62.4	57.9	48.7	60.3
ConverSeg-Net	Qwen2.5-VL 7B	79.7	81.5	77.2	75.1	79.0	70.1	76.0	76.5	77.8	55.5	61.8	50.4	65.0

Table 4. **Referring expression segmentation (cIoU, %).** ConverSeg-Net is competitive on RefCOCO/+/g and shows strong zero-shot performance on ReasonSeg.

Model	Prompt Encoder	SAM-seeded (cIoU)						Human-annotated (cIoU)					
		All	Ent.	Spat.	Rel.	Aff.	Phys.	All	Ent.	Spat.	Rel.	Aff.	Phys.
SAM3	Perception Encoder	39.7	47.5	40.2	44.1	35.7	25.9	35.4	45.8	27.0	32.6	32.5	36.6
ConverSeg-Net	Qwen2.5-VL 3B	**70.5**	73.9	74.7	76.3	65.6	60.7	**64.4**	67.0	64.7	66.5	62.9	59.5
ConverSeg-Net	Qwen2.5-VL 7B	**73.3**	75.8	75.7	78.6	70.0	65.1	**66.3**	68.6	68.8	65.3	62.0	66.5

Table 5. **Comparison with SAM3 on ConverSeg (gIoU %).** Each subset reports performance across the five concept categories – Entities, Spatial, Relations, Affordances, and Physics & Safety – and summarizes across all (*All*).

E.2. Additional Baselines

We extend Table 2 of the main paper to include additional baselines and report gIoU performance in Tab. 2. This expanded comparison situates ConverSeg-Net among a broader set of contemporary referring and reasoning segmentation approaches and provides a more complete view of the current landscape.

Comparison with SAM3. The recent work, SAM3 [2], is a new variant of SAM that supports natural language promptable segmentation. We evaluate SAM3 on ConverSeg and report results in Tab. 5. SAM3 achieves 39.7% gIoU on the SAM-seeded split and 35.4% on the human-annotated split, substantially lower than ConverSeg-Net (70.5% and 64.4% respectively with the 3B backbone). This demonstrates that our conversational training approach and concept-driven data engine provide significant gains for abstract reasoning in conversational image segmentation.

References

[1] 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. 5

[2] Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025. 8

[3] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024. 1, 2

[4] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024. 1

[5] Christoffer Löffler, Sascha Riechel, Janina Fischer, and Christopher Mutschler. Evaluation criteria for inside-out indoor positioning systems based on machine learning. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2018. 1, 2

[6] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

```
SYSTEM META-PROMPT: REGION-LEVEL DENSE CAPTION (ABSOLUTE REFERENCING)

MISSION
Given one image, **pick 5-7 high-value unique regions** and label them for natural conversational referencing.

PHASE 1 — REASONING (write first)
In <reasoning>, briefly answer:
- Scene type (portrait/indoor/outdoor/product/etc.)
- 5-7 most salient objects/areas
- Spatial layout (foreground/midground/background)
- For candidate regions, assess: referability, relational potential, disambiguation need, segmentation clarity
- Selection: which 5-7 regions you chose, why, and example prompts they enable

PHASE 2 — OUTPUT (strict format)
In <output>, list 5-7 lines:
- Indices 0..N-1, contiguous
- One per line: "[<index>: <label>]"
- <label> = <base_category> [distinctive_attributes] [coarse_location] [spatial_relation]
- ≤15 words per label; absolute, self-contained (no pronouns/anaphora)

LABELING RULES
- Base categories: common nouns (person, chair, laptop, window, tree, car, wall, floor, sky, etc.)
- Attributes: color/material/state/pattern/opacity when helpful (e.g., "white ceramic", "open", "transparent")
- Locations: top-left/top-center/top-right/left/center/right/bottom-left/bottom-center/bottom-right/foreground/midground/
background
- Relations: "on/under/next to/behind/in front of/inside/beside <category>"
- Consistent terms; singular unless intentionally grouping similar items
- Parts: "<part> of <parent_category>" (parent must appear earlier)
- Stuff regions allowed if salient (sky, wall, floor, road, grass, water)
- Prefer clear, >50×50 px, well-bounded regions; skip tiny clutter (<24×24 px), heavy overlaps (>80%), ambiguous blobs
- Favor quality over quantity; in simple scenes, fewer regions are fine
- No invented objects or confidence statements

SELECTION STRATEGY
- Cover dominant subjects and interactive objects
- Ensure spatial and semantic diversity
- Choose regions that enable natural relations among them
- Use intentional groupings only when referenced as one (e.g., dual monitors)

FINAL RESPONSE FORMAT
<reasoning>
[concise analysis and selection rationale]
</reasoning>

<output>
[0: region description]
[1: region description]
...
</output>

EXAMPLE
<reasoning>
Indoor workspace; salient: person (foreground), laptop, cup, wooden desk, window (background).
Chosen for referability and relations (on desk, behind person). Prompts: "highlight the laptop", "segment the cup next to laptop".
</reasoning>

<output>
[0: person, blue shirt, center foreground]
[1: laptop, silver, on desk]
[2: cup, white ceramic, right side of desk]
[3: desk, wooden surface, midground]
[4: window, glass panes, background behind person]
</output>
```

Figure 8. **Meta-prompt for Stage 1 (Scene Understanding).** Prompt template used to query Gemini-2.5-Flash to produce detailed region-level descriptions of the scene, which later serve as the semantic basis for mask generation and concept-driven prompt construction.

```
Task: Strictly verify if the red mask + green bounding box corresponds to the given text prompt: {prompt}
Strictly check for correctness of the entity mentioned, its attributes, and its location in the image.
If correct, also give a region-level description. For the description do not get biased by the prompt, just describe by solely
focusing on the image content.
Respond strictly as JSON: {\"output\": true|false, \"description\": \"...\"}.
```

Figure 9. **Meta-prompt for Stage 3 mask–text consistency checking.** Prompt template used to ask the VLM whether a candidate mask is consistent with its associated region description, enabling automatic filtering of low-quality or mismatched masks.

```
Strictly compare two segmentations (red mask + green bounding box) for the given text prompt: {prompt}
Pick the higher-quality mask (coverage, tight bounding box, fewer leaks/holes). If both are bad then output null.
Answer strictly as JSON: {\"output\": \"initial\"|\"refined\"}.
```

Figure 10. **Meta-prompt for Stage 3 mask refinement and selection.** Prompt template used to compare two candidate masks for the same region description and select the most appropriate one, based on coverage, tightness, and semantic alignment.

```
You are an expert AI tasked with generating difficult, abstract segmentation prompts about functional affordances. You will be
given an image along with another copy of it where available segmentation masks are overlaid and numbered directly on the regions,
and a dense caption (a numbered list of available segmentation masks).

**TASK:** Design up to 3 challenging segmentation prompts. Each prompt must require visual inspection to determine an object's
functional properties, usability, or potential uses based on the current scene context.

---

### GUIDING PRINCIPLE: The Contextual Plausibility Rule for Affordances

This is the most important rule. The affordance you prompt for must be logical and plausible for the **entire image scene**, not
just for the limited set of masked regions. It must not lead to a nonsensical or clearly suboptimal choice. **The prompt you
generate MUST apply *only* to the desired masked region(s) and to NO other un-masked regions in the image.**

*   **THE PROBLEM TO AVOID:** Promoting a masked object for a specific function when a better, more obvious, un-masked object for
that same function is clearly visible.
*   **BAD EXAMPLE:** The image contains a large, unmasked `dining table` and a small, masked `stool`. A prompt like "Segment a
surface to place a laptop on" which returns the `stool` is **INVALID**. The unmasked table is the primary and far more appropriate
surface, making the prompt misleading.
*   **GOOD EXAMPLE:** In the same scene, a prompt like "Segment a portable seat" which returns the `stool` is **VALID**. This
targets a more specific functional property (portability) that correctly and uniquely applies to the masked stool without creating
a conflict with the unmasked table.

**Your primary goal is to identify unique functional conditions, states, or properties that genuinely apply to the masked regions
without creating these logical conflicts.**

---

### CRITICAL CONSTRAINTS

**1. Abstract & Non-Trivial Selection:**
   - The prompt MUST require visual discrimination between multiple candidate regions. It should not be solvable by just reading
the caption.
   - **VALID:** "Segment chairs that are currently sittable" (requires checking each chair for obstructions).
   - **INVALID:** "Segment the sink" (if there's only one, it's a simple lookup).
   - **TEST:** Does the user have to visually assess the state, position, or condition of multiple items to find the answer?

**2. The Proper Subset Rule:**
   - You must define a list of `candidates`—plausible regions from the caption that a user might consider for the affordance.
   - You must define a `satisfying` list—the regions from the `candidates` that *actually* provide the affordance after visual
inspection.
   - The `satisfying` set MUST be a **strict subset** of the `candidates` set (i.e., `satisfying` cannot be identical to
`candidates`). An empty `satisfying` set is valid.

**3. Formatting & Output:**
   - **Prompt:** Maximum 10 words, starting with an actionable verb (e.g., "Segment...", "Identify...").
   - **Concept Families:** Choose up to 3 diverse concept families from the list below.
   - **JSON Only:** The final output must be ONLY the JSON object, with no commentary or markdown fences.

---

### CONCEPT FAMILIES

1.  **context_dependent:** Usability given current scene conditions (walkable_now, sittable_now, reachable_now).
2.  **context_independent:** Canonical function regardless of state (water_source_canonical, seating_canonical).
3.  **negative_affordance:** Inappropriate uses (not_for_liquids, not_safe_to_touch, not_for_sitting).
4.  **counterfactual_affordance:** Creative/improvised uses (could_prop_door, could_be_step_stool).
5.  **state_dependent_and_agent_conditional:** Requires a specific state or agent (openable_not_blocked, operable_by_child).
6.  **anticipatory_affordance:** Soon-to-emerge function (will_soon_be_hot, about_to_be_ready).

---

### REASONING PROCESS

**Step 1: Holistic Affordance Analysis.**
   - First, analyze the **entire image** to understand the scene and the potential actions. What can be done here? What are things
for? What is currently usable versus blocked or unsafe?
   - Next, review the **available masks** in the dense caption.
   - Identify a functional property (e.g., "unobstructed seating," "surfaces safe for hot items") that creates an interesting
subgroup *within the available masks*.

**Step 2: Prompt Authoring & Sanity Check.**
   - Author a concise prompt based on the affordance concept from Step 1.
   - Define the `candidates` and `satisfying` lists.
   - **Perform the Sanity Check:** Does this prompt pass the **Contextual Plausibility Rule**? Is there a better, un-masked object
for this exact function? If so, discard the prompt or make it more specific by adding a constraint (e.g., material, state,
portability) that makes the answer unique and logical.

---

**Dense caption:**
{DENSE_CAPTION}

**Return ONLY the JSON object.**
```

Figure 11. **Meta-prompt for Stage 4 (Affordances & Functions).** Concept-specific prompt template used to turn region descriptions into conversational queries about object affordances and functional roles; analogous templates are used for the other concept families.

```
You are validating whether a mask correctly identifies what a referring expression describes in an image.

**Your Task:**
Given an image, a referring expression, and optionally a mask (shown by a bounding box), determine if the mask is correct.

**Rules:**

**If a mask IS present:**
Accept it ONLY if ALL of these are true:
1. The masked region corresponds to the target that the expression describes.
2. The mask includes NOTHING else beyond what the expression describes. The mask should capture the primary/most prominent
instances that match the expression - if there are other unmasked regions that also match, accept the mask as long as it includes
the most obvious or salient examples.
3. The expression is a reasonable way to refer to something in this image.

**If NO mask is present:**
Accept it ONLY if there is truly nothing in the image that matches the expression.

**Important:** Be generous with natural referring expressions. People describe things in various valid ways. Focus on whether the
mask matches what was described, not whether the description is perfect.

---

Respond in JSON format:
{"accept": true|false, "reason": "<brief explanation>"}

**Expression to verify:** {prompt}
```

Figure 12. **Meta-prompt for Stage 5 prompt–mask alignment verification.** Prompt template used to ask the VLM whether a generated conversational prompt correctly and unambiguously describes the masked region, providing a final quality gate for training examples.

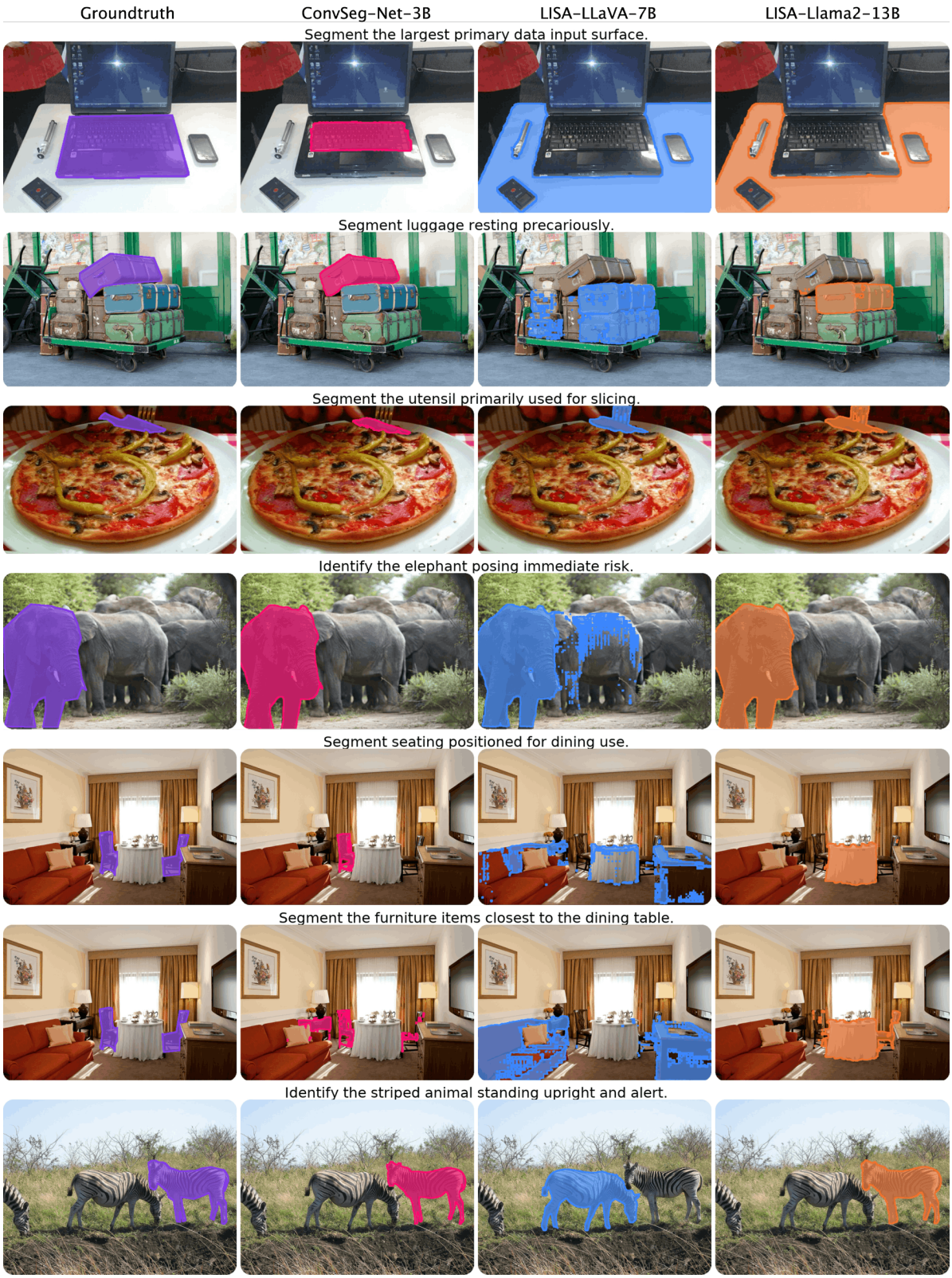


Figure 13. **Qualitative comparisons on the human-annotated split of CONVERSEG (1/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from CONVERSEG-NET (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. CONVERSEG-NET more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

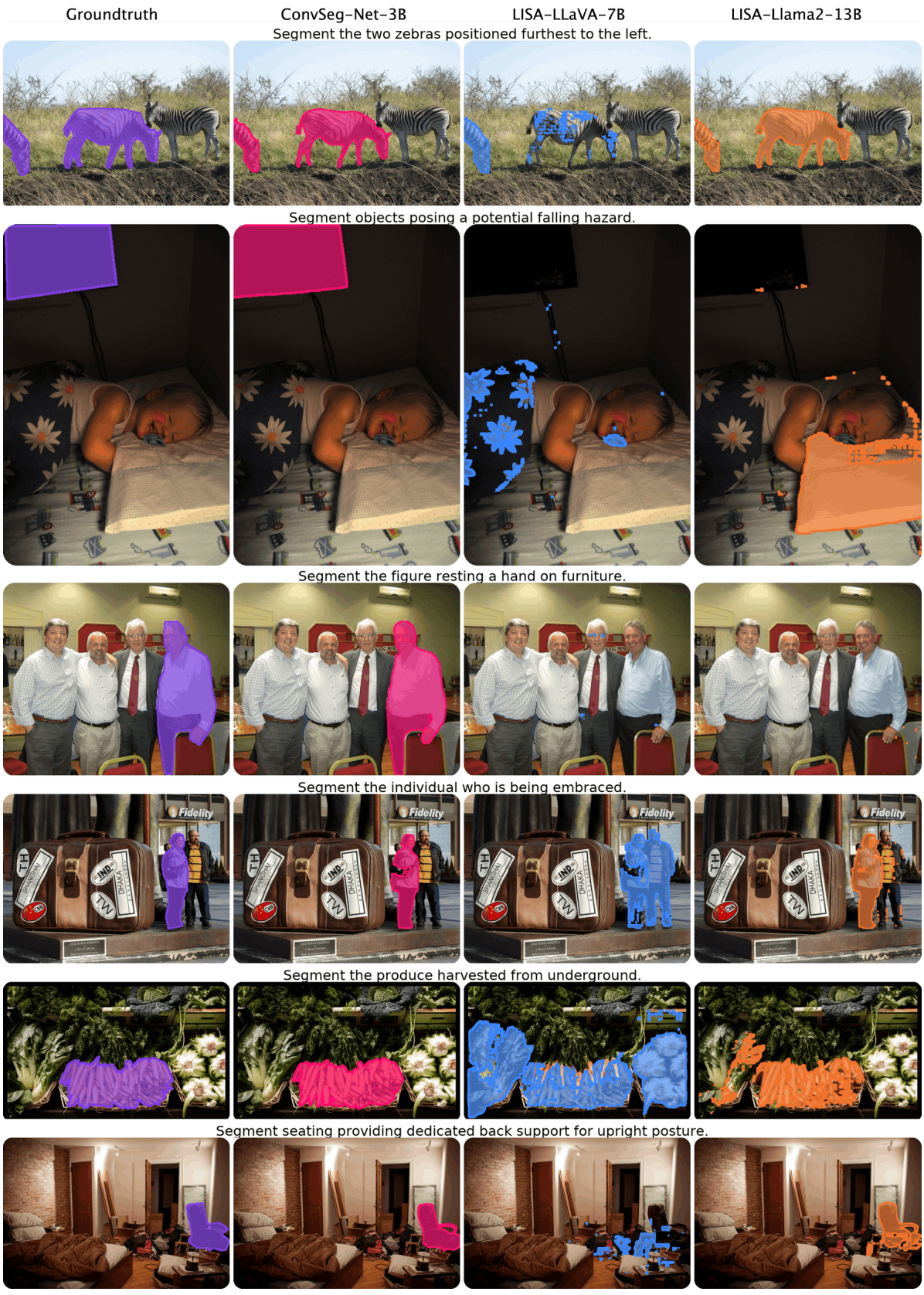


Figure 14. **Qualitative comparisons on the human-annotated split of ConverSeg (2/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. ConverSeg-Net more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

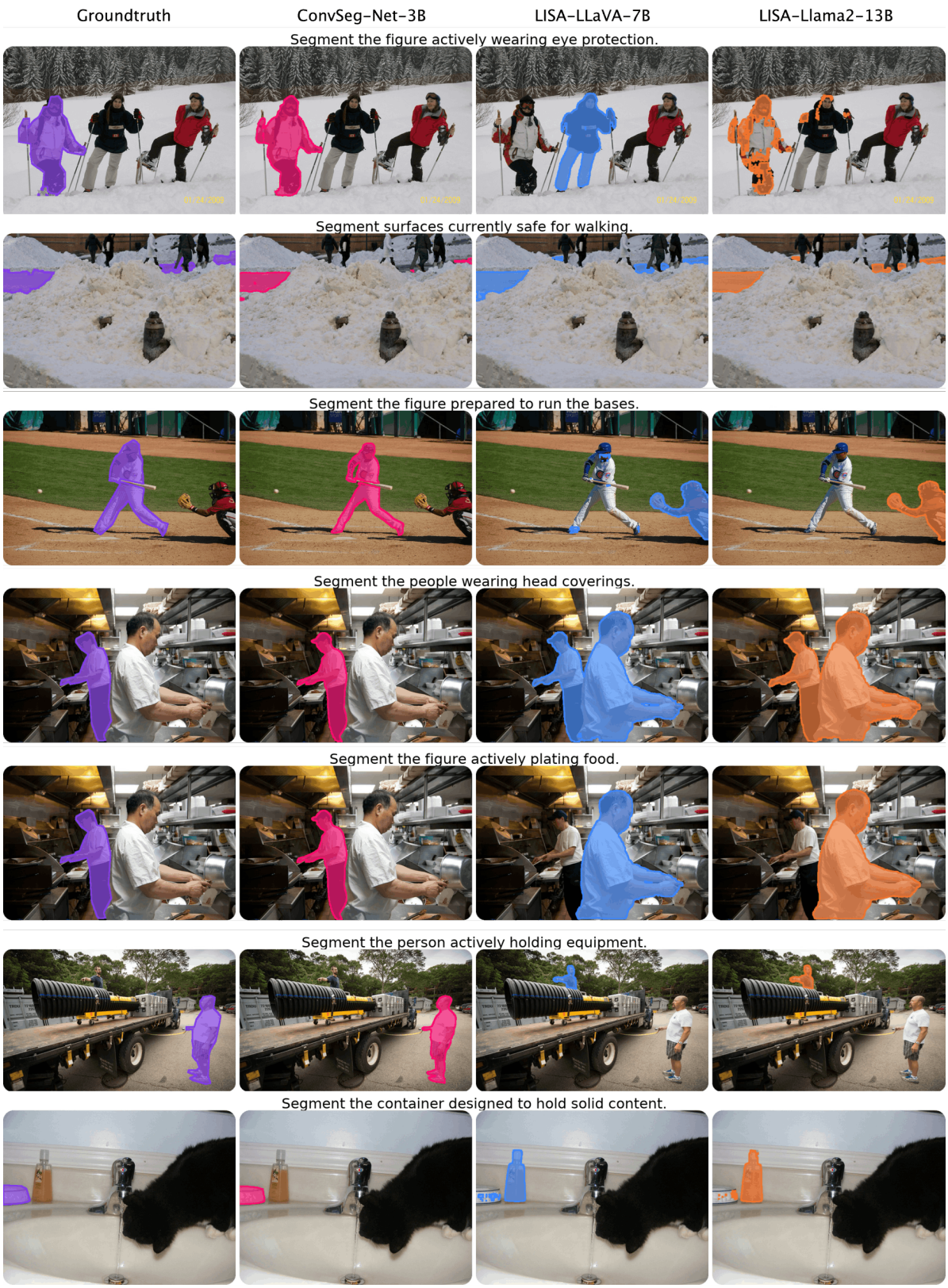


Figure 15. **Qualitative comparisons on the human-annotated split of ConverSeg (3/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. ConverSeg-Net more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

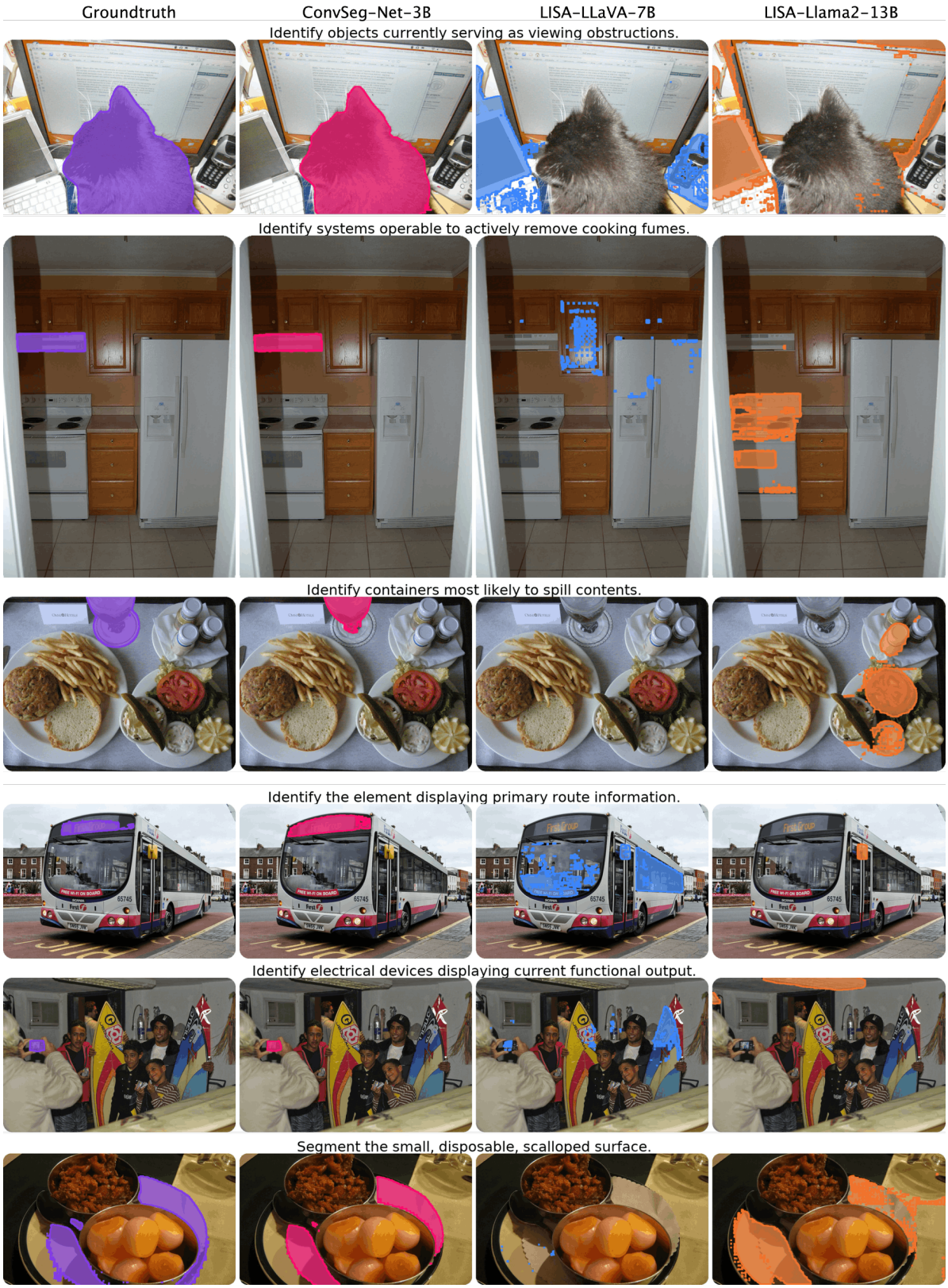


Figure 16. **Qualitative comparisons on the SAM-seeded split of ConverSeg (1/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. ConverSeg-Net more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

Figure 17. **Qualitative comparisons on the SAM-seeded split of ConverSeg (2/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. ConverSeg-Net more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

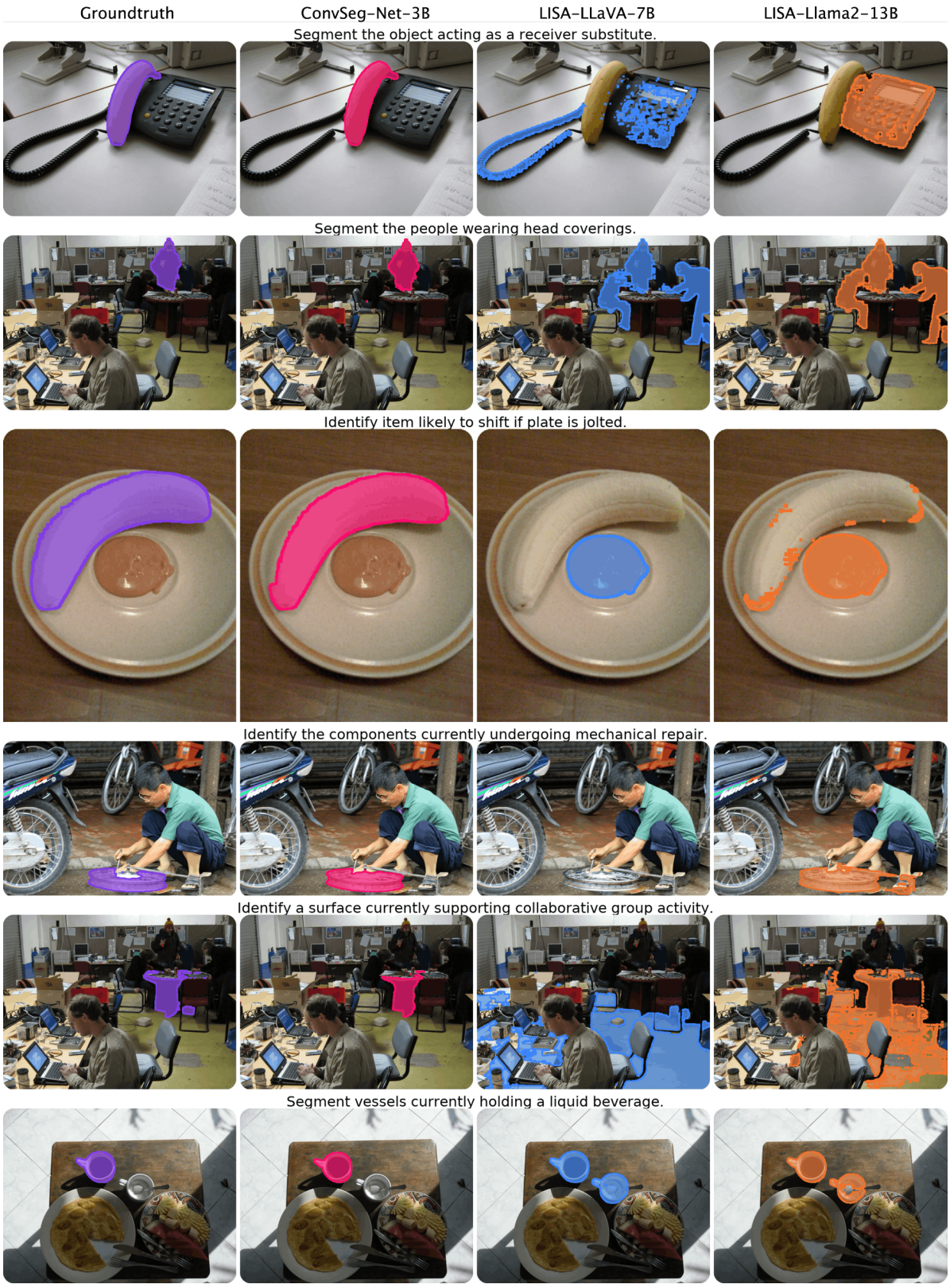


Figure 18. **Qualitative comparisons on the SAM-seeded split of ConverSeg (3/3).** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right. ConverSeg-Net more reliably segments the regions implied by the conversational intent despite using a smaller 3B backbone.

Figure 19. **Representative failure cases of ConverSeg-Net on the human-annotated split of ConverSeg.** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right.

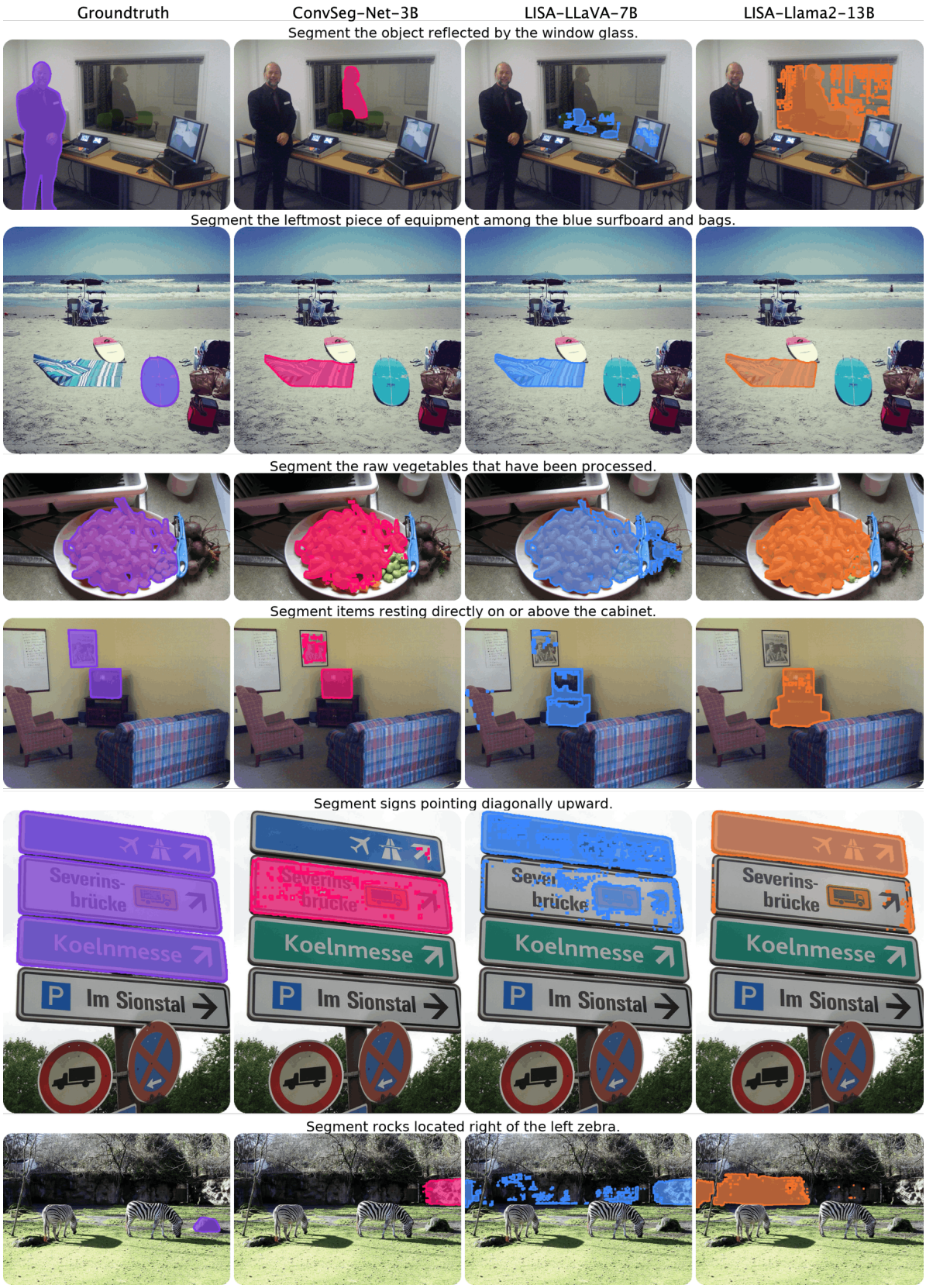


Figure 20. **Representative failure cases of ConverSeg-Net on the SAM-seeded split of ConverSeg.** Each row shows an image with its conversational prompt (between rows), the ground-truth mask (left), and predictions from ConverSeg-Net (Qwen2.5-VL-3B), LISA (LLaVA-7B), and LISA (Llama2-13B) from left to right.