RobotSeg: A Model and Dataset for Segmenting Robots in Image and Video (*Supplementary Material*)

Haiyang Mei Qiming Huang Hai Ci Mike Zheng Shou*

Show Lab, National University of Singapore

https://github.com/showlab/RobotSeg

1. Overview

In this supplementary material, we provide additional experiments, analyses, and implementation details to complement the main paper. We first present an application example of RobotSeg for robot-centric data augmentation in Section 2. We then show more dataset examples from our VRS benchmark in Section 3. Next, we report comprehensive category-wise evaluations across robot embodiments in Section 4, highlighting the robustness of RobotSeg across diverse robot embodiments. We include more visual comparisons with state-of-the-art methods in Section 5 and provide computational efficiency analyses in Section 6. We further provide an additional comparison with SAM 3 [3] for completeness in Section 7. Finally, detailed architecture descriptions of key modules, including the memory encoder, the structure perceiver, and the mask decoder, are provided in Section 8, followed by discussions on limitations and future directions in Section 9.

2. An Application Example of RobotSeg

Accurate robot segmentation is a prerequisite for reliable robot perception and tracking, and it further enables the creation of high-quality training data for robot learning. A practical example is robot-centric data augmentation, where robot masks are used to composite the robot into diverse scenes to improve the robustness of downstream policies. However, this process is highly sensitive to mask quality: segmentation errors such as missing parts, drifted boundaries, or broken structures directly translate into unrealistic composites that can harm the learning signal.

Figure 1 illustrates this effect on video frames. RoboEngine [8], which operates on individual images, often produces fragmented or structurally damaged masks (Figure 1c), while SAM 2.1 [7] requires manual clicks and still struggles to maintain spatial and temporal consistency across frames (Figure 1d). These inaccuracies propagate to the augmented images, leading to visually implausible robot placements with missing limbs or distorted geometry (Figure 1f-g).

In contrast, RobotSeg generates clean, complete, and temporally stable masks (Figure 1e), enabling high-fidelity robot compositing (Figure 1h). The preserved geometry and consistent boundaries result in realistic augmented images that maintain the structural integrity of the robot. This example highlights the practical value of precise robot segmentation in creating large-scale, diverse, and reliable training data for robot learning systems.

3. More VRS Dataset Examples

To complement the dataset overview in the main paper, we present additional examples from our video robot segmentation (VRS) dataset in Figure 2 and 3. These examples further illustrate the diversity of robot embodiments, manipulation behaviors, and scene contexts captured in VRS. Each example shows the RGB video frames (top) and their corresponding annotation masks (bottom), where the robot arm and gripper are labeled separately following the hierarchical labeling protocol.

Across these samples, VRS demonstrates substantial variation in motion patterns, viewpoints, backgrounds, object interactions, and lighting conditions. Such diversity is essential for training and evaluating models that aim to achieve robust, temporally consistent robot segmentation in realistic and dynamic environments. By providing continuous video sequences with fine-grained arm and gripper masks, VRS enables research on temporal modeling, mask propagation, and structure-aware segmentation beyond what is possible with the image-only dataset RoboEngine [8].

4. Category-Wise Analysis Across Diverse Robot Embodiments

To further evaluate the robustness of different models under diverse robot embodiments, we conduct a category-wise

* Corresponding Author

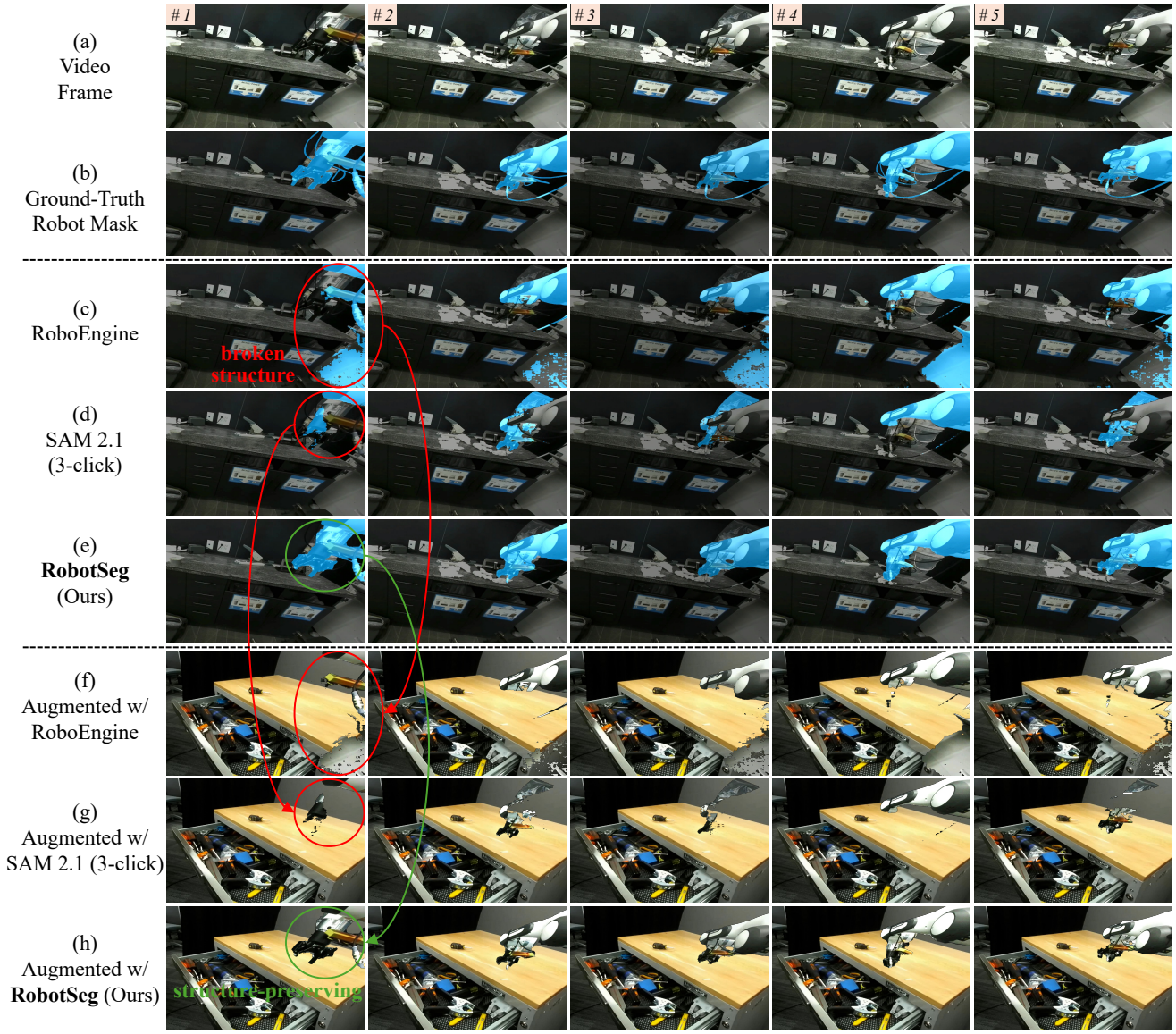

Figure 1. Robot segmentations obtained with the image robot segmentation method RoboEngine [8] (c) and the promptable video segmentation method SAM 2.1 (with 3 manual clicks to initialize the segmentation) [7] (d), compared to our RobotSeg model (e). When used for robot data augmentation, inaccurate masks from RoboEngine and SAM 2.1 lead to broken or unrealistic robot composites (f-g), whereas our RobotSeg enables clean and structurally accurate augmentation (h) by precisely preserving the robot regions.

analysis on the VRS dataset. While the main manuscript reports overall results for whole-robot, robot-arm, and robot-gripper segmentation, such aggregated metrics may conceal substantial variations across robots with distinct structures, kinematics, and visual characteristics. Therefore, we additionally present per-category results for all 10 robot embodiments, separately for whole robot (Table 1), robot arm (Table 2), and robot gripper (Table 3).

Across the three segmentation targets, RobotSeg consistently ranks among the top-performing methods within each robot embodiment. RobotSeg ranks first in 7, 8, and 6 out of 10 robot types for segmenting the whole robot (Table 1), robot arm (Table 2), and robot gripper (Table 3), respectively. These results indicate that RobotSeg maintains stable performance across diverse robot embodiments with different shapes, sizes, and visual characteristics. In contrast, existing approaches, including the ***robot-specific methods*** RoVi-Aug [2] and RoboEngine [8], the ***language-conditioned models*** CLIPSeg [6], LISA [5], EVF-SAM [9], and VideoLISA [1], the ***promptable video foundation model*** SAM 2.1 [7] (with one manual click prompt to initialize the segmentation), and the ***concept seg-***

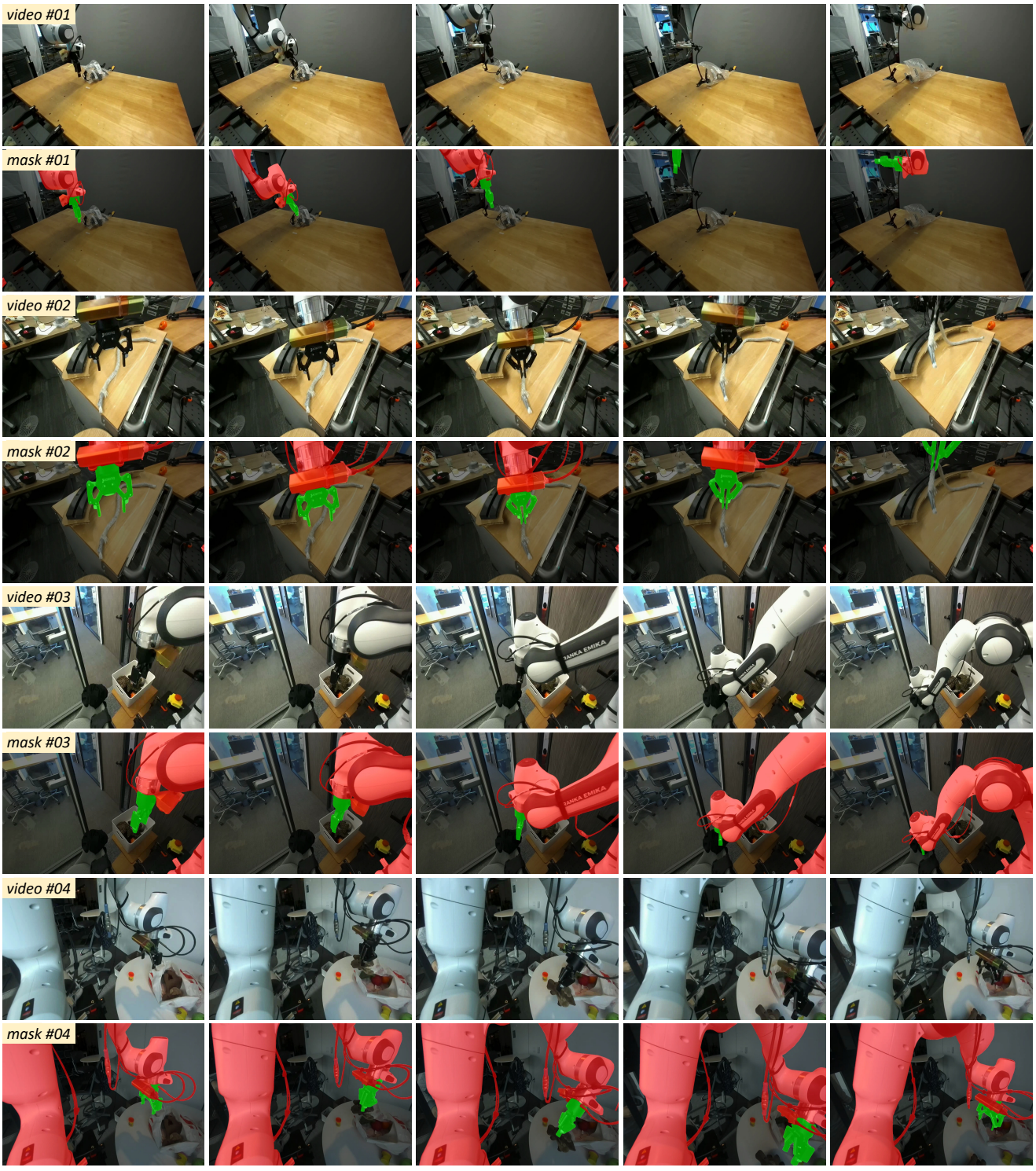

Figure 2. Examples from our video robot segmentation (VRS) dataset. Each example shows the RGB sequence (top) and robot annotation masks (bottom), where the robot arm is highlighted in red and the gripper in green.

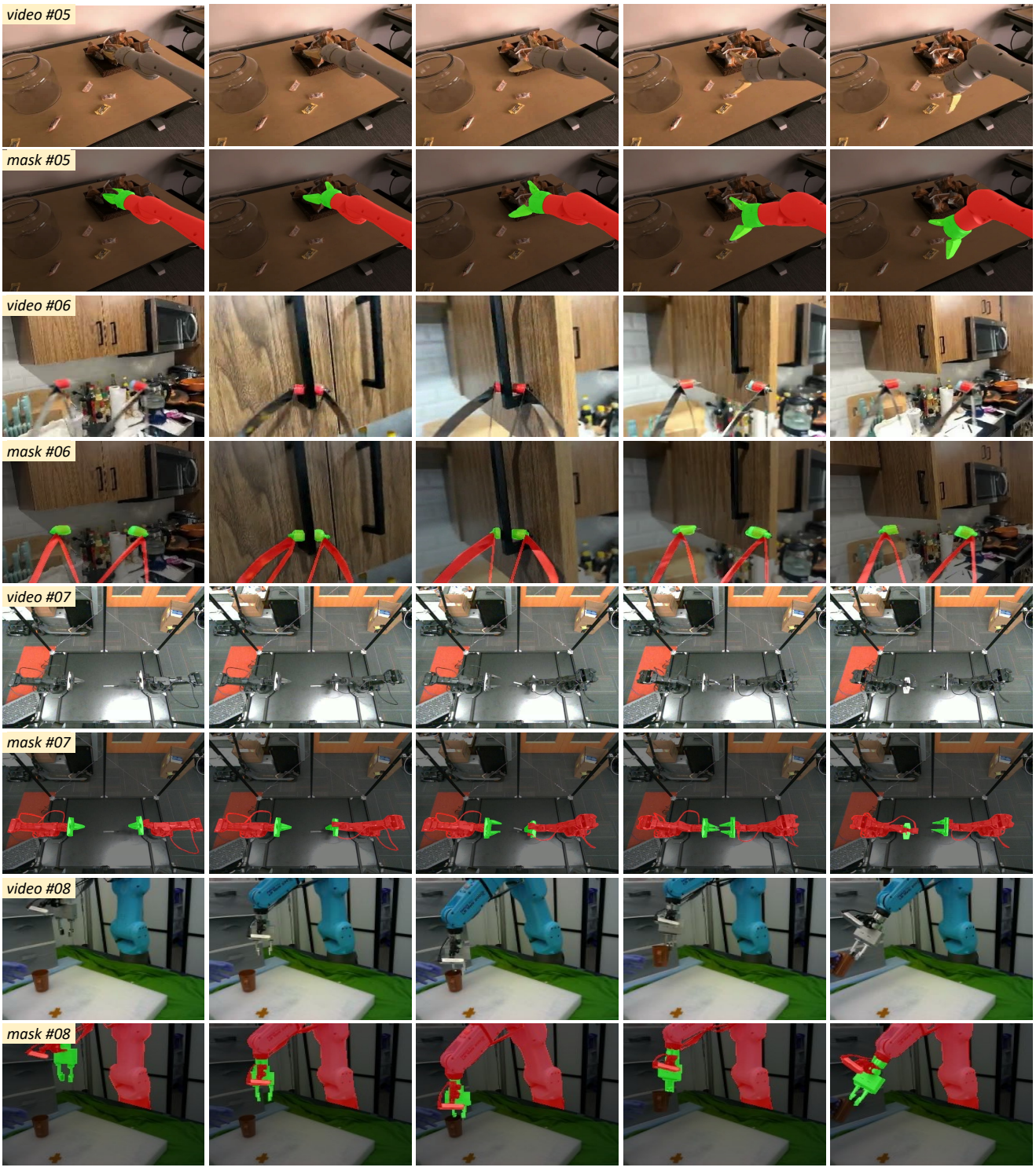

Figure 3. Examples from our video robot segmentation (VRS) dataset. Each example shows the RGB sequence (top) and robot annotation masks (bottom), where the robot arm is highlighted in red and the gripper in green.

Table 1. Comparison of "whole robot" segmentation across the 10 robot categories in the VRS dataset. The superscript $(1c)$ denotes that SAM 2.1 [7] uses a manual click for initialization.

Methods	Para. (M)	#01 Franka	#02 Fanuc Mate	#03 UR5	#04 Kuka iiwa	#05 Google Robot	#06 MobileALOHA	#07 xArm	#08 WindowX	#09 Sawyer	#10 Hello Stretch	**Overall**
RoVi-Aug [2]	638.5	29.8	69.9	51.1	72.2	34.6	41.6	81.1	49.1	41.7	19.1	38.9
RoboEngine [8]	898.4	75.8	**88.5**	78.2	87.8	89.8	73.4	94.0	**91.4**	83.2	6.4	74.1
CLIPSeg [6]	150.8	20.6	38.6	32.6	59.5	33.1	29.1	52.3	29.5	35.4	15.0	26.6
LISA [5]	13992.9	47.1	48.6	80.0	80.5	78.3	51.8	88.6	74.4	55.4	35.2	55.3
EVF-SAM [9]	898.4	56.7	56.4	81.0	68.7	82.6	61.2	93.0	81.3	58.9	52.0	63.9
VideoLISA [1]	4788.3	48.9	58.8	67.5	75.3	75.9	63.4	93.5	46.4	57.2	35.7	53.6
SAM 2.1$^{(1c)}$ [7] (Original)	39.0	25.0	35.6	78.8	8.8	64.7	21.1	30.3	88.0	22.2	14.6	38.2
SAM 2.1$^{(1c)}$ [7] (Finetuned)	39.0	74.7	81.4	80.4	**91.4**	78.8	45.6	91.0	85.4	87.5	68.5	73.6
SAM 3 [3]	860.1	30.5	59.9	0.0	70.8	19.0	41.5	92.7	60.2	8.4	0.0	34.7
RobotSeg (Ours)	41.3	**82.1**	85.0	**88.0**	85.3	**90.2**	**87.1**	**95.6**	87.5	**95.3**	**78.7**	**85.1**

Table 2. Comparison of "robot arm" segmentation across the 10 robot categories in the VRS dataset. The superscript $(1c)$ denotes that SAM 2.1 [7] uses a manual click for initialization.

Methods	Para. (M)	#01 Franka	#02 Fanuc Mate	#03 UR5	#04 Kuka iiwa	#05 Google Robot	#06 MobileALOHA	#07 xArm	#08 WindowX	#09 Sawyer	#10 Hello Stretch	**Overall**
CLIPSeg [6]	150.8	23.3	45.4	32.8	65.7	22.7	26.2	44.3	10.7	22.1	19.7	23.7
LISA [5]	13992.9	45.8	57.0	62.1	75.2	54.4	39.6	90.9	24.9	45.5	25.1	42.1
EVF-SAM [9]	898.4	51.3	56.2	66.2	76.4	56.5	38.6	91.5	27.5	41.1	29.3	44.7
VideoLISA [1]	4788.3	39.5	45.1	58.5	64.5	54.9	43.2	89.9	26.5	37.5	48.5	41.4
SAM 2.1$^{(1c)}$ [7] (Original)	39.0	26.3	41.4	52.9	9.1	58.4	16.4	31.7	30.6	26.9	18.7	28.9
SAM 2.1$^{(1c)}$ [7] (Finetuned)	39.0	**79.0**	79.0	81.2	**87.1**	60.5	38.7	87.2	54.5	72.4	62.5	66.2
SAM 3 [3]	860.1	51.9	62.4	74.0	69.6	64.3	52.2	89.4	26.8	39.4	0.0	45.0
RobotSeg (Ours)	41.3	76.6	**79.8**	**83.4**	82.9	**85.6**	**72.2**	**92.4**	**61.2**	**95.0**	**76.0**	**75.6**

mentation model SAM 3 [3], show large variation across robot categories. The category-wise analysis indicates that RobotSeg generalizes more reliably across real-world robotic embodiments than existing models.

These results highlight two key insights. First, robot segmentation is highly embodiment-dependent: strong average performance does not guarantee robustness across distinct robot embodiments, as illustrated by SAM 2.1 (finetuned) whose robot-arm accuracy is 66.2 overall but drops to 38.7 on the MobileALOHA robot (Table 2). Second, RobotSeg's robot-aware design and label-efficient video training enable generalization across diverse embodiments for whole-robot, arm-level, and gripper-level segmentation. This demonstrates that RobotSeg is not only favorable on average but also reliably transferable to a wide spectrum of real-world robotic embodiments.

5. More Visual Comparison Results

Figure 4, 5, and 6 provide additional qualitative comparisons under the automatic segmentation setting, covering the three levels of robot granularity: whole robot, robot arm, and robot gripper. These examples complement the limited visualizations included in the main paper due to space constraints and further highlight the challenges posed by diverse embodiments and cluttered scenes. We also include prompt-based comparisons in Figure 7 and 8, where segmentation is guided by a single click or a bounding box on the first video frame. Together, these qualitative results provide a comprehensive view of segmentation performance under both automatic and prompt-guided modes.

Under the ***automatic segmentation*** setting, RoboEngine [8] exhibits clear inaccuracies and temporal inconsistencies for the whole robot segmentation (Figure 4). In the top example, it incorrectly segments background clothing as part of the robot in the second and fourth columns, while only loosely capturing the robot region in the third column. Similar issues appear throughout the sequence. For the robot arm segmentation (Figure 5), RoboEngine cannot distinguish the arm from the gripper, so we instead compare our RobotSeg with EVF-SAM [9], which achieves the best per-

Table 3. Comparison of "robot gripper" segmentation across the 10 robot categories in the VRS dataset. The superscript $(1c)$ denotes that SAM 2.1 [7] uses a manual click for initialization.

Methods	Para. (M)	#01 Franka	#02 Fanuc Mate	#03 UR5	#04 Kuka iiwa	#05 Google Robot	#06 MobileALOHA	#07 xArm	#08 WindowX	#09 Sawyer	#10 Hello Stretch	**Overall**
CLIPSeg [6]	150.8	2.4	8.3	1.9	0.9	6.7	10.1	2.9	15.2	2.7	5.8	6.7
LISA [5]	13992.9	9.3	7.0	26.4	6.8	35.1	29.2	2.9	43.5	33.1	12.9	21.2
EVF-SAM [9]	898.4	11.4	27.0	22.6	10.1	33.8	34.0	10.9	38.8	38.6	15.0	23.8
VideoLISA [1]	4788.3	4.5	2.2	19.0	1.5	34.3	29.7	4.4	29.4	27.4	8.0	15.9
SAM 2.1$^{(1c)}$ [7] (Original)	39.0	52.0	35.6	42.1	21.8	46.7	15.3	**88.0**	72.1	48.0	48.3	47.7
SAM 2.1$^{(1c)}$ [7] (Finetuned)	39.0	52.8	**70.0**	53.8	**66.8**	59.6	70.3	18.6	79.3	76.1	**78.6**	64.8
SAM 3 [3]	860.1	11.9	0.5	9.3	5.8	25.0	11.6	10.0	5.9	17.0	5.6	10.5
RobotSeg (Ours)	41.3	**71.6**	62.4	**68.2**	64.6	**80.7**	**78.7**	78.8	**85.0**	**87.5**	77.4	**76.0**

formance among language-conditioned methods. In the top example, EVF-SAM identifies only the robot base while entirely missing the articulated arm. In the bottom example, it mistakenly labels a coffee machine with similar color as the robot and produces temporally unstable predictions. In the robot gripper segmentation (Figure 6), EVF-SAM again fails to localize the correct component: in the top example, it segments the arm instead of the gripper, and in the bottom example, the final two columns incorrectly segment clothing as the gripper. In contrast, our RobotSeg consistently produces accurate, component-specific masks with stable temporal behavior across diverse embodiments and complex backgrounds, demonstrating strong robustness in the automatic segmentation setting.

Figure 7 and 8 show comparisons with SAM 2.1 [7] under ***prompt-based segmentation***. In the 1-click setting (Figure 7), a single point is provided on the first video frame (green star). In the top example, SAM 2.1 only segments a partial portion of the robot, while in the bottom example, it confuses the black robot gripper with the similarly colored background, leading to incomplete or mixed masks. RobotSeg, however, generates complete and clean robot masks without confusing foreground and background. In the bounding-box setting (Figure 8), where a box is given on the first video frame (green rectangle), SAM 2.1 again shows inconsistency: the top example under-segments the robot, while the bottom example over-segments into surrounding regions. In contrast, our RobotSeg delivers stable and temporally consistent results in both sequences. Overall, the prompt-based comparisons confirm that our RobotSeg remains robust and accurate when initialized with minimal user input.

Across all automatic and prompt-guided settings, these visual comparisons collectively demonstrate that RobotSeg provides accurate, temporally consistent, and embodiment-robust robot segmentation, significantly outperforming existing methods under challenging real-world scenarios.

Table 4. Comparison of the computational efficiency of different methods. For each method, we list model parameters, FLOPs, and average inference time per frame.

Methods	Para. (M)	FLOPs (G)	Time (ms)
RoVi-Aug [2]	638.5	546.1	87.9
RoboEngine [8]	898.4	1753.4	250.7
CLIPSeg [6]	150.8	98.5	95.3
LISA [5]	13992.9	61820.4	431.3
EVF-SAM [9]	898.4	1753.4	250.7
VideoLISA [1]	4788.3	36300.6	670.8
SAM 2.1 [7]	39.0	284.3	68.6
SAM 3 [3]	860.1	5045.2	160.1
RobotSeg (Ours)	41.3	319.8	94.2

6. Computational Efficiency Analysis

We focus in this work on developing RobotSeg, the first foundation model for robot segmentation that supports both images and videos. To provide a complete view of its practical applicability, we report computational efficiency comparisons in Table 4.

From Table 4, RobotSeg offers a clear computational advantage over existing robot segmentation baselines. Its overall FLOPs (319.8G) are substantially lower than robot-specific methods such as RoVi-Aug [2] and RoboEngine [8], as well as language-conditioned approaches including LISA [5], EVF-SAM [9], and VideoLISA [1], all of which rely on considerably larger backbones and therefore incur much higher computational cost. Although CLIPSeg [6] is lightweight in terms of FLOPs, its significantly lower segmentation performance renders it unsuitable for precise robot segmentation. Compared with SAM 2.1 [7] (284.3G), RobotSeg introduces only a modest increase in FLOPs while providing capabilities that SAM 2.1 does not support: (i) fully automatic robot segmentation without requiring manual prompts to initialize segmentation, and (ii) substantially improved segmenta-

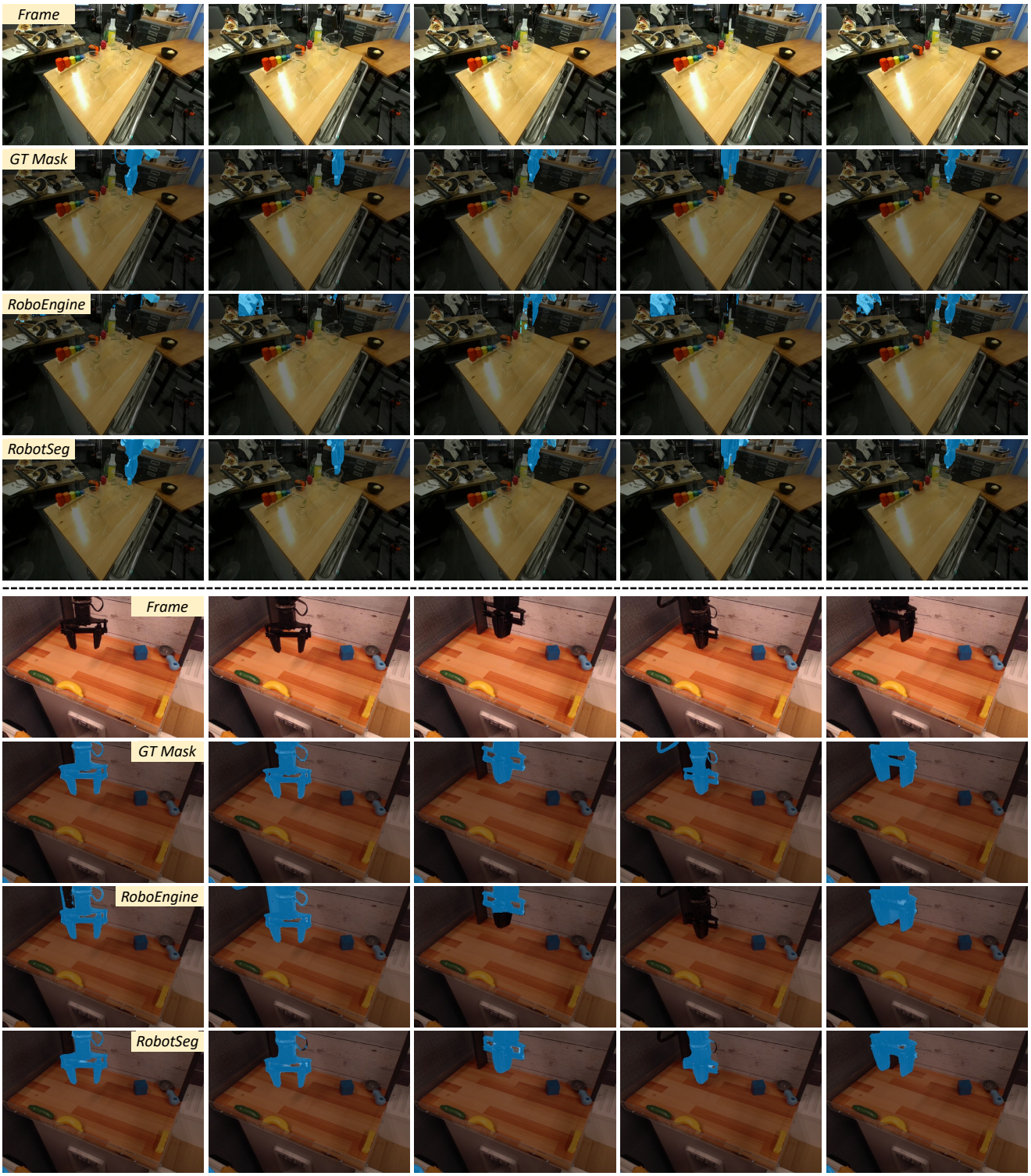

Figure 4. Qualitative comparison of whole-robot segmentation under the automatic setting. RoboEngine [8] exhibits clear inaccuracies and temporal inconsistencies, while our RobotSeg produces accurate and stable masks across frames.

Figure 5. Qualitative comparison of robot arm segmentation under the automatic setting. EVF-SAM [9] struggles to localize the articulated arm and often confuses background objects, whereas our RobotSeg provides accurate, component-specific, and temporally consistent predictions.

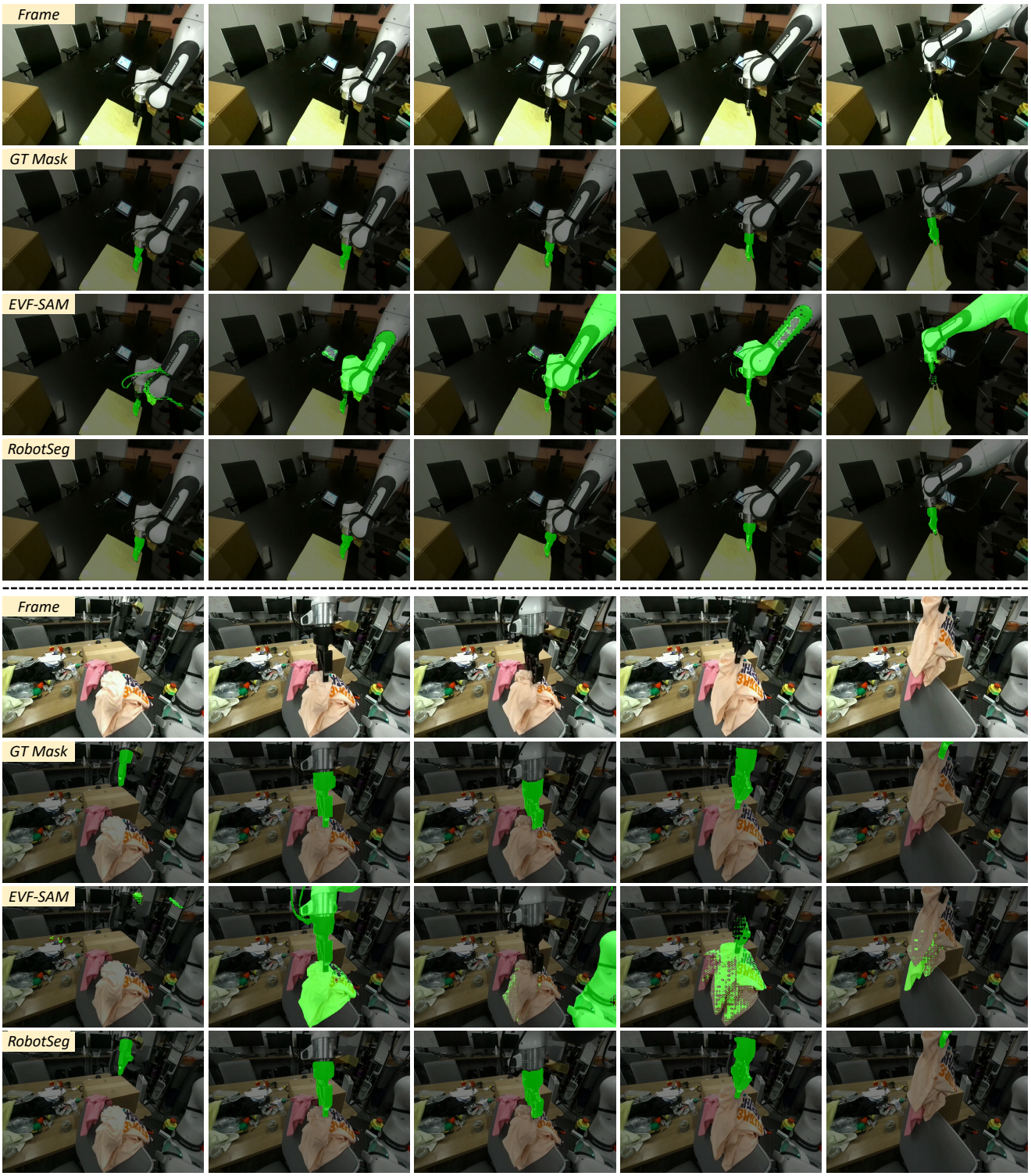

Figure 6. Qualitative comparison of robot gripper segmentation under the automatic setting. EVF-SAM [9] frequently misidentifies the gripper or mistakes background regions for the target, while our RobotSeg consistently segments the correct component with stable temporal behavior.

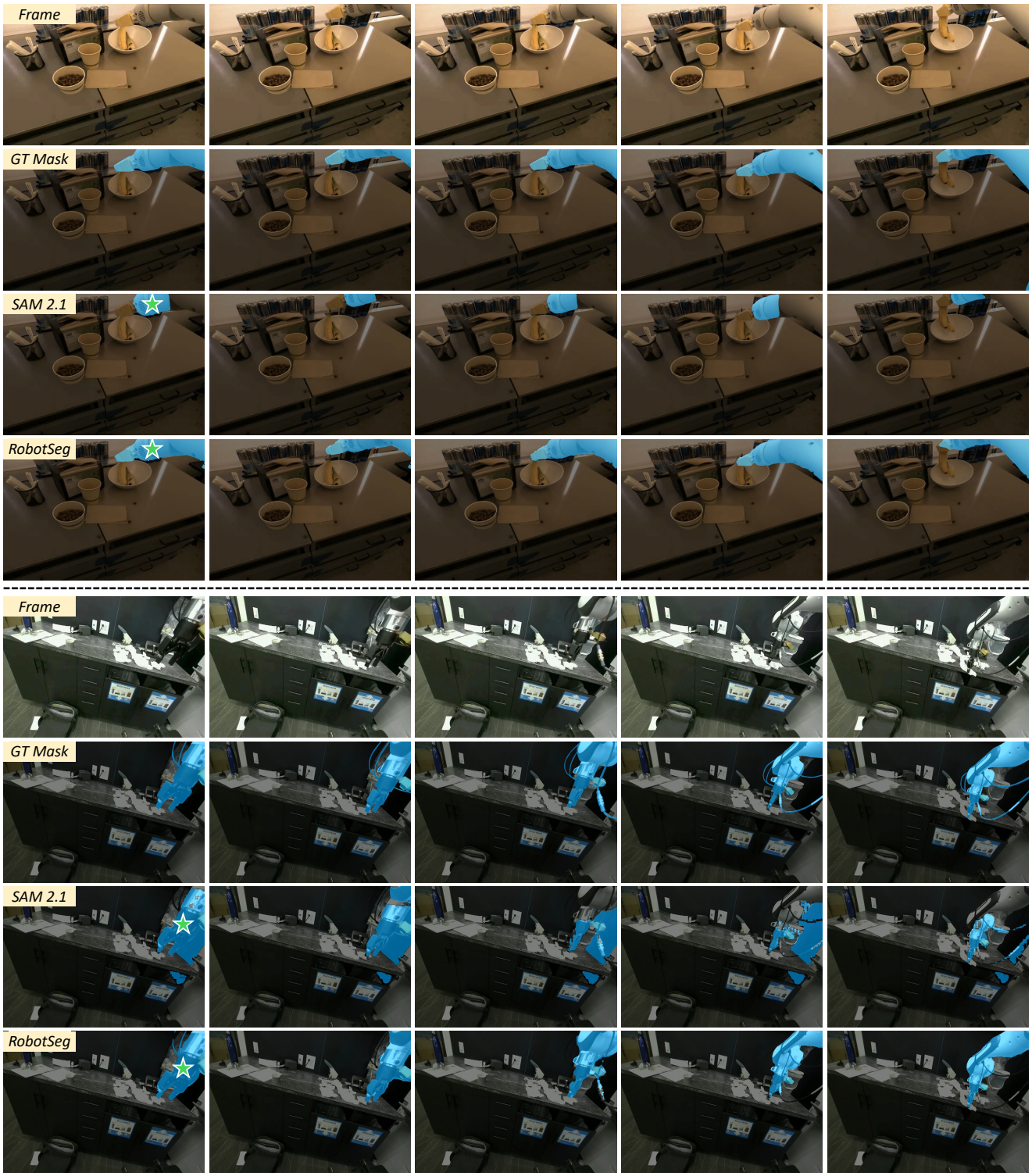

Figure 7. Comparison of robot segmentation results when a single click (green star) is given on the first video frame. SAM 2.1 [7] often produces incomplete masks or confuses dark grippers with the background, whereas our RobotSeg generates complete and clean segmentation across frames.

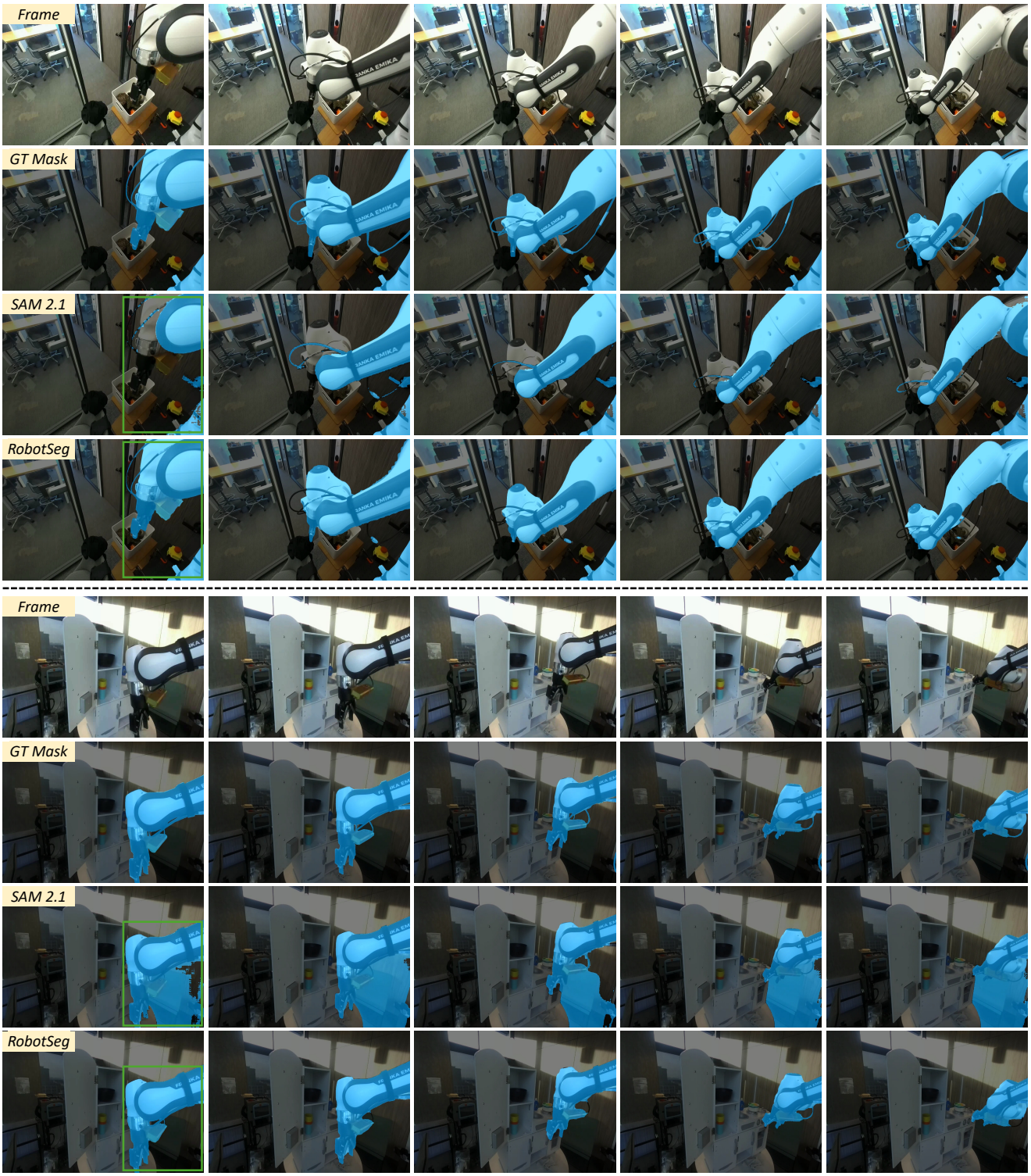

Figure 8. Comparison of robot segmentation results when a bounding box (green rectangle) is provided on the first video frame. SAM 2.1 [7] shows inconsistent behavior, including under- and over-segmentation, while our RobotSeg maintains accurate and temporally consistent predictions.

tion quality across both images and videos. Moreover, our RobotSeg achieves an average inference time of 94.2 ms per frame (>10 FPS) (measured on an NVIDIA RTX A5000 GPU), which remains competitive among high-capacity models and supports practical deployment. Overall, these results show that RobotSeg preserves computational efficiency while offering much stronger task-specific performance, making it a practical choice for both academic research and real-world deployment.

7. Comparison with SAM 3

SAM 3 [3] is a concept segmentation model that supports noun-phrase prompts. Although it is designed for open-vocabulary concept segmentation rather than robot-specific perception, it can be directly applied to our task by prompting it with phrases such as "robot", "robot arm", or "robot gripper".

We apply SAM 3 to our VRS benchmark using concept prompts corresponding to robot components. As summarized in Table 1,2,3, SAM 3 is indeed capable of segmenting robots at the concept level, but its performance remains noticeably below that of RobotSeg. SAM 3 is not tailored for articulated robot geometry, fine-grained arm–gripper separation, or temporally consistent segmentation in complex robot manipulation videos, which limits its performance on these scenarios. In contrast, RobotSeg incorporates structure-enhanced memory association and robot-specific prompt generation, achieving substantially higher accuracy and stability.

Overall, these results indicate that while SAM 3 [3] provides a general open-vocabulary interface that can segment robots via concept prompts, dedicated modeling is required to achieve high-quality robot segmentation in realistic and dynamic environments.

8. Architecture Details

This section provides extended architectural details that complement the model description in the main manuscript.

8.1. Memory Encoder

The memory encoder is responsible for transforming the predicted masks and image encoder embeddings into a compact representation that can effectively guide subsequent frames. As illustrated in Figure 10 (a), the memory encoder first processes the previous-frame mask through a downsampling branch and projects the previous image feature into a consistent embedding space. These two sources are then fused via lightweight convolutional operations to integrate both spatial cues and semantic context. The fused representation is subsequently passed through an output projection layer, producing the memory feature used for temporal guidance in the following frames.

8.2. Structure Perceiver

The structure perceiver uses the memory feature produced by the memory encoder to guide the extraction of robot-aware structural cues and the generation of the structure map. As illustrated in Figure 10 (b), the edge-enhanced features are processed by a multi-scale depthwise convolution module (3×3, 5×5, and 7×7) to capture edge patterns at different receptive fields. The aggregated multi-scale features are then aligned with the memory feature through a cross-attention layer, enabling the model to inject temporal robot context into the structural perceiving. Finally, a 3×3 convolution predicts the structure map, which serves as structure guidance for refining following representations.

8.3. Mask Decoder

The mask decoder is responsible for generating segmentation masks conditioned on image embeddings and prompt tokens. Our implementation extends the SAM mask decoder by introducing additional robot prompts, enabling automatic and more consistent robot segmentation across video frames.

In Figure 9, the prompt tokens represent optional user guidance such as clicks or bounding boxes, while the robot tokens encode robot semantic and temporal context extracted from previous frames, guiding the segmentation in the current frame. The inclusion of robot tokens enhances temporal consistency by leveraging historical robot information. The mask decoder uses a sequence of two-way transformer blocks that perform bidirectional attention between image embeddings and token representations. Specifically, the following attention mechanisms are used:

1. **Self-attention**: applied to tokens to learn interactions within the token space.
2. **Token-to-image attention**: enables token embeddings to query relevant image features.
3. **Image-to-token attention**: aggregates token responses into the image feature representations.

After feature fusion through the transformer blocks, the decoder outputs multiple candidate masks to handle prompt ambiguities. Following SAM 2 [7], we retain only the mask with the highest predicted Intersection-over-Union (IoU) score for propagation. Additionally, the decoder includes an occlusion prediction head implemented as an MLP, which predicts whether the target robot is visible in the current frame. This capability is important for accurately handling partial or full occlusions of the robot.

9. Limitation and Future Work

While our RobotSeg demonstrates strong overall performance across diverse embodiments and challenging scenes, several limitations remain. First, although RobotSeg consistently outperforms existing models, it does not achieve the

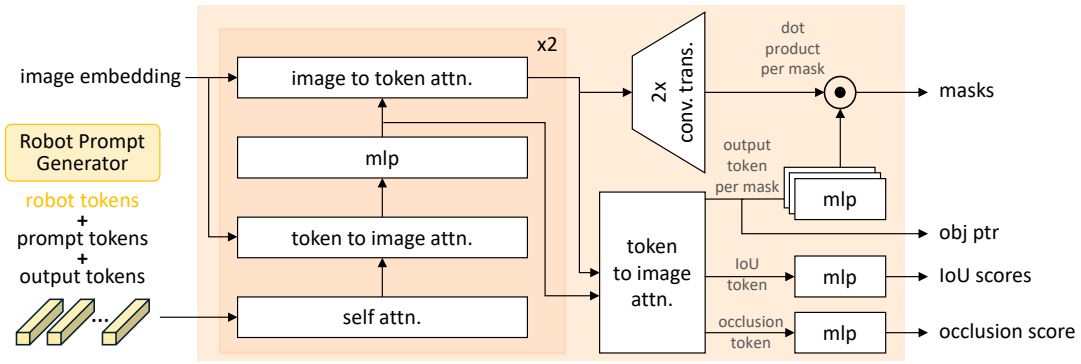

Figure 9. Architecture details of the mask decoder.

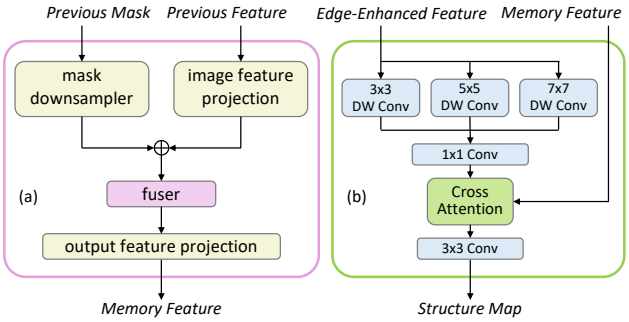

Figure 10. Architecture details of (a) the memory encoder and (b) the structure perceiver.

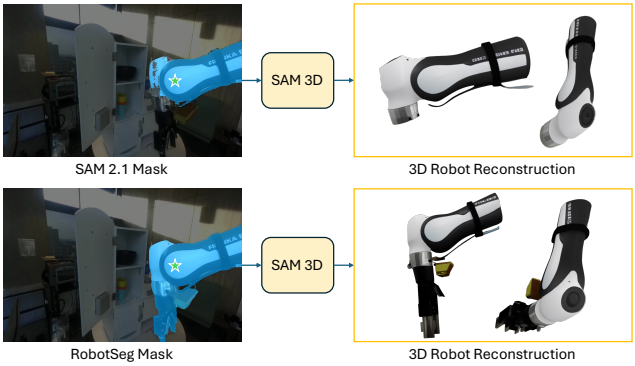

Figure 11. Compared with the incomplete robot mask from SAM 2.1 [7] (top row), our RobotSeg provides the complete and accurate robot mask (bottom row), enabling SAM 3D [4] to generate accurate and faithful 3D robot reconstructions.

highest accuracy on every individual robot category. Certain robots with unusual appearances or scene conditions remain challenging, suggesting that further embodiment-specific modeling could provide additional gains. Second, the current design introduces moderate computational overhead compared to the original SAM 2.1. Although RobotSeg remains efficient relative to existing baselines, there is still room for reducing FLOPs and model parameters, especially for deployment on resource-constrained platforms such as mobile manipulators or embedded robotic systems.

These limitations open several promising directions for future work. One direction is to explicitly incorporate additional modalities such as depth, motion cues, or tactile signals, which may help disambiguate difficult cases where RGB appearance is insufficient. Another direction is to develop more lightweight architectures or distillation strategies that retain RobotSeg's robustness while significantly reducing computational cost. Finally, integrating RobotSeg into closed-loop robotic systems and studying its impact on downstream tasks such as 3D robot reconstruction (*e.g.*, Figure 11), policy learning, manipulation, and navigation represents an exciting avenue for advancing robot perception and control.

References

[1] Zechen Bai, Tong He, Haiyang Mei, Pichao Wang, Ziteng Gao, Joya Chen, Zheng Zhang, and Mike Zheng Shou. One token to seg them all: Language instructed reasoning segmentation in videos. *NeurIPS*, 37:6833–6859, 2024. 2, 5, 6

[2] Lawrence Yunliang Chen, Chenfeng Xu, Karthik Dharmarajan, Richard Cheng, Kurt Keutzer, Masayoshi Tomizuka, Quan Vuong, and Ken Goldberg. Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning. In *Conference on Robot Learning*, pages 209–233, 2025. 2, 5, 6

[3] Meta Superintelligence Labs. Sam 3: Segment anything with concepts, 2025. Accessed: 2025-11-20. 1, 5, 6, 12

[4] Meta Superintelligence Labs. Sam 3d: 3dfy anything in images, 2025. Accessed: 2025-11-20. 13

[5] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *CVPR*, pages 9579–9589, 2024. 2, 5, 6

[6] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *CVPR*, pages 7086–7096, 2022. 2, 5, 6

[7] 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. In *ICLR*, 2025. 1, 2, 5, 6, 10, 11, 12, 13

[8] Chengbo Yuan, Suraj Joshi, Shaoting Zhu, Hang Su, Hang Zhao, and Yang Gao. Roboengine: Plug-and-play robot data augmentation with semantic robot segmentation and background generation. *IROS*, 2025. 1, 2, 5, 6, 7

[9] Yuxuan Zhang, Tianheng Cheng, Lianghui Zhu, Rui Hu, Lei Liu, Heng Liu, Longjin Ran, Xiaoxin Chen, Wenyu Liu, and Xinggang Wang. Evf-sam: Early vision-language fusion for text-prompted segment anything model. *arXiv:2406.20076*, 2024. 2, 5, 6, 8, 9