

M³-VOS: Multi-Phase, Multi-Transition, and Multi-Scenery Video Object Segmentation

Supplementary Material

Overview

We introduce:

- More implementation details of our work in Secs. 1 to 7.
- More experiments about the challenge in M³-VOS in Secs. 8 to 10.
- More failure cases in Sec. 11.

1. Details of Annotations

1.1. Phase Definition

We list the specific definitions of phase below:

- **Solid:** Volume is relatively fixed, has distinct boundaries, and shapes independent of the container.
 - **Particulate:** Composed of several fragmented parts.
 - **Non-particulate:** Composed of single/few larger parts.
 - * **Rigid Body:** Exhibiting a relatively fixed shape and resistance to deformation.
 - * **Flexible Body:** Has a relatively unstable shape and can undergo deformation easily.
- **Liquid:** Volume is relatively fixed and has distinct boundaries, fluidity, or shape dependent on the container.
 - **Viscous Fluid:** Has significant viscosity, and can stretch.
 - **Non-viscous Liquid:** No significant viscosity, cannot stretch.
- **Aerosol/Gas:** Volume not fixed, has no distinct boundaries, shape dependent on the container.

1.2. Phase Transition Definition

In our works, we ensure that the definition of phase transitions meets a fundamental requirement: The transition from an initial state to a final state may correspond to different specific phase transitions depending on the characteristics of the transformation. However, for a specific phase transition, its initial and final states must be unique.

We list all the initial and final states for each phase transition as Tab. 1. Besides, in Tab. 2, we give a detailed definition of different phase transitions.

2. Connected Component Jaccard Index

To avoid ignorance of the small part during evaluation, we introduce the connect component Jaccard Index \mathcal{J}_{cc} . The definition of \mathcal{J}_{cc} is the average Jaccard Index of the maximum bipartite matching corresponding to all connected

mask components between the ground truth and the predicted image.

We implemented our \mathcal{J}_{cc} using the *Hungarian* algorithm, different from the one in the official implementation of VSCOS [5] that calculates the \mathcal{J}_{cc} using a two-loop matching process, *i.e.*, iteratively finding for each connected component in Mask A the one in Mask B that maximizes the Jaccard Index.

3. Details of Masks SQA

3.1. Three Criteria in Masks SQA

We design three criteria to evaluate the annotation in M³-VOS, including:

- **Tracking Accuracy**
 - 0: Target is lost or tracked incorrectly for a long time.
 - 1: Target is lost or tracked incorrectly for a short continuous period.
 - 2: Target is lost or tracked incorrectly in a few isolated frames.
 - **3: Target is always tracked correctly.**
- **Mask Annotation Completeness**
 - 0: Mask has been completely missing for a long time.
 - 1: Mask has been partially missing for a long time.
 - 2: Mask is partially missing in some frames.
 - **3: Mask is complete and accurate throughout.**
- **Mask Boundary Stability**
 - 0: Mask boundary shows an obvious jitter for a long time.
 - 1: Mask boundary shows a slight jitter for a long time.
 - 2: Mask boundary shows a slight jitter for a short time.
 - **3: Mask boundary shows no visible jitter.**

3.2. SQA Analyze

We select three experienced reviewers to evaluate all of our masks in M³-VOS in the criteria of Sec. 3.1 using the reviewer UI as Fig. 1. In the process of constructing M³-VOS, we make sure the scores in any criterion of all of our masks are higher than 2. In the final evaluation of M³-VOS, the MOS in these criteria are 2.95 in tracking accuracy, 2.91 in mask annotation completeness, and 2.89 in mask boundary stability.

4. Details of Avoidance of Model Bias

In this part, we introduce the details of the dual-model cross-validation method. In this process, we validate that

Table 1. The different phase transitions and the unique initial state and final state. We give some examples to highlight their characteristic.

Category	Phase Transition	Initial State	Final State	Example
Intra-Phase (Solid)	Separate	Rigid	Rigid	<i>Disassembling a gun, Taking apart building blocks</i>
	Twist	Flexible	Flexible	<i>Knead dough, Tie shoelaces</i>
	Break	Rigid	Particulate	<i>Break cups, Chop vegetables</i>
	Stretch	Flexible	Flexible	<i>Pull noodles, Pull rubber</i>
	Split	Particulate	Particulate	<i>Sift the rice, Sieve the sand</i>
	Merge	Rigid	Rigid	<i>assemble guns, Jigsaw puzzle</i>
	Crush	Rigid	Particulate	<i>Grind the herb, Crush the stone</i>
Intra-Phase (Liquid)	Flow	Non-Viscous	Non-Viscous	<i>Pour Water, Pour tea</i>
	Paint	Liquid	Liquid	<i>Paint the wall, Paint in oil</i>
	Splash	Non-Viscous	Non-Viscous	<i>Diving sports, Cast a stone into the water</i>
	Mix	Non-Viscous	Non-Viscous	<i>Milk pouring art, Paint mixing with water</i>
	Drip	Non-Viscous	Non-Viscous	<i>Drip the acid, Drip the eye drops</i>
Intra-Phase (Aerosol/Gas)	Diffuse	Aerosol/Gas	Aerosol/Gas	<i>Smoke spreads, Mist spreads</i>
Cross-Phase	Solidify	Liquid	Solid	<i>Water freezes, Chocolate hardens into solid chocolate</i>
	Melt	Solid	Liquid	<i>Melt chocolate, Melt the ice</i>
	Deposition	Aerosol/Gas	Solid	<i>Form dew, Condense into alcohol</i>
	Vaporize	Liquid	Aerosol/Gas	<i>Humidifier sprays water, Boil water</i>
	Crystallize	Liquid	Solid	<i>Making salt, Making sugar</i>
	Sublimate	Solid	Aerosol/Gas	<i>Burn coal, Burn plastic</i>
	Dissolve	Solid	Liquid	<i>Dissolve the tablet, Make formula</i>
	Compress	Solid	Liquid	<i>Juicing fruits, Extracting pomegranate juice</i>
	Flow out	Solid	Non-Viscous	<i>Break chocolate with a liquid center</i>
	Soften	Solid	Viscous	<i>Boil sugar, Bake cheese</i>

our dataset pipeline efficiently declines the model bias of annotations.

4.1. IoU Analysis

In terms of model selection of the dual-model cross-validation process, we adapt the annotation model to the latest SAM2 [3]. We utilized the open-source base plus model configuration and checkpoints, as this configuration is more effective in fully segmenting our target objects compared to other model setups.

In the dual-model cross-validation, we first randomly sampled a subset of videos annotated by Cutie at a ratio of 5:1. We selected 6 volunteers from a total of 12 to re-annotate this subset using both the SAM2-assisted and Cutie-assisted annotation tools, resulting in masks designated as Mask A and Mask B, respectively. To balance annotation efficiency and validation effectiveness, we set the annotation frame rate to 6 fps in the cross-validation. The

high-frame-rate annotated masks obtained for the dataset are referred to as Mask O.

By calculating the Intersection over Union (IoU) and the other two metrics introduced in [4, 5], the results are shown in Fig. 2, indicated that J_{st} (MaskA, MaskB) and J_{mean} exceeded 85% and were very close to each other. Specifically, although the difference between Mask A and Mask B is slightly larger than that between Mask B and Mask O, we have Eq. (1) holds:

$$\mathcal{J}_{\sigma}(B, O) - \mathcal{J}_{\sigma}(A, O) \ll 1 - \mathcal{J}_{\sigma}(B, O). \quad (1)$$

These results suggest that the annotations SAM2-assisted annotation tool produced are comparable to those of the Cutie-assisted tool, without significant bias due to model differences. They also indicate that the bias introduced by the models can be considered negligible compared to other sources of systematic error, such as volunteer annotation habits and inadvertent jitters during annotation.

Table 2. The detail of the definition of different phase transitions.

Category	Phase Transition	Definition
Intra-Phase (Solid)	Separate	Block-like solid objects are disassembled into multiple block-like pieces
	Twist	Flexible objects are deformed into various shapes.
	Break	Solid objects are shattered into countless small fragments.
	Stretch	Flexible objects are elongated into a longer form.
	Split	The solid particles disperse in all directions, spreading out from the source.
	Merge	Multiple block-like objects are combined into a single whole.
	Crush	Solid block-like objects are ground into powdery granules.
Intra-Phase (Liquid)	Flow	The liquid moves as a whole under the influence of external force.
	Paint	The liquid is applied onto a solid surface.
	Splash	The liquid is scattered in all directions due to a sudden external force.
	Mix	One liquid is poured into another, causing the two liquids to blend together.
	Drip	A small amount of liquid is transferred drop by drop.
Intra-Phase (Aerosol/Gas)	Diffuse	The gas or aerosol spreads out, gradually expanding its presence in the air.
Cross-Phase	Solidify	The liquid turns into a solid as it cools or hardens.
	Melt	The solid turns into a highly fluid liquid.
	Deposition	The gas directly transforms into a solid without passing through the liquid state.
	Vaporize	The liquid turns into a gas as it heats up and evaporates.
	Crystallize	The solid crystals form and separate out from the liquid.
	Sublimate	The solid directly produces gas as it transitions without becoming a liquid.
	Dissolve	The substance disperses evenly in the liquid, forming a solution.
	Compress	The solid is squeezed under pressure, forcing a large amount of liquid to be released.
	Flow out	The liquid content flows out from within the solid as it is released or displaced.
	Soften	The solid gradually turns into a thick, viscous liquid.

4.2. Blind Review: DMOS Evaluation

In addition to the quantitative analysis of model bias conducted using the dual-model validation, we also designed a mechanism for blind comparison by experienced reviewers. We presented 3 reviewers with both Mask A and Mask B from the cross-validation annotation process, allowing them to evaluate the performance of the two masks based on the three criteria mentioned in Sec. 3.1. The reviewers were instructed to select the mask they deemed superior. If they considered the performances to be equivalent, they could choose *Equal*. Throughout this process, the order of Mask A and Mask B was randomized to ensure that the reviewers were unaware of which mask corresponded to which model.

The final subjective evaluation results are shown in Fig. 3, indicating that the two masks demonstrated a considerable degree of consistency across the three subjective evaluation metrics, with no significant bias observed.

5. Multi-Level Semi-Auto Annotation Tool

In Fig. 4, we show the details of the interactive UI of the multi-level semi-auto annotate tool. We implement this tool based on the interactive demo from Cutie [1], including pixel level, appearance-level, and object level. In particular, we implement the object-level function using the SAM2 model and Cutie model. In this way, we could perform the dual-model cross-validation analysis in Sec. 4.

6. Details of Core Subset

We extract a subset of cases that better represents the full dataset and refer to it as the core subset. For each specific scenario, we extracted a subset of cases. During the selection of the core subset in each scenario, we consider a series of factors: the number of the full set, the number of classes included, and the difficulty of cases. As is shown in Tab. 3, we choose the size of the core subset of each scenario to

Table 3. Details of core subset number of different Scenarios

Scenario	Full Set Number	Core Subset Number	Class Number	Example
Factory	67	9	13	Disassemble/assemble a gun, Wrap a wire
Handicraft	40	12	14	Knit a sweater, Wrap a cigar
Kitchen	163	12	70	Cut celery, Shave fish
Lab	152	12	56	Drip liquid, Dissolve drug
Housework	3	3	2	Twist mop
Decoration	9	2	2	Tear wallpaper
Hospital	7	2	4	Ground herbal
School	1	1	1	Sharpen a pencil
Farm	13	3	7	Shear a sheep
Sport	2	1	1	Hit a balloon
Daily live	45	14	18	Pour tea, Shave beard
Experiment field	14	6	3	Break glass, Twist a rubber

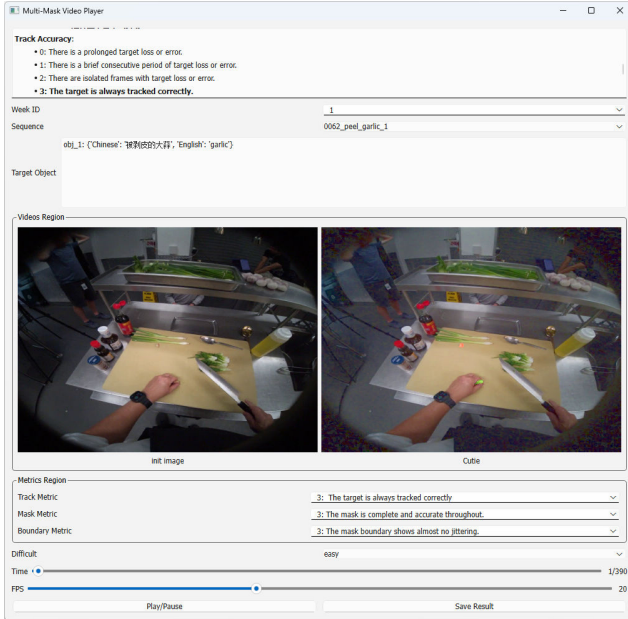


Figure 1. Review UI. The reviewer is required to evaluate the mask annotation from three criteria.

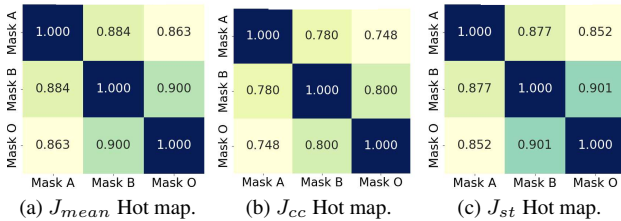


Figure 2. IOU Analysis among Mask A (SAM2), Mask B (Cutie Dual), and Mask O (Cutie in the final dataset).

make it closer to the proportion of the full set and the class number.

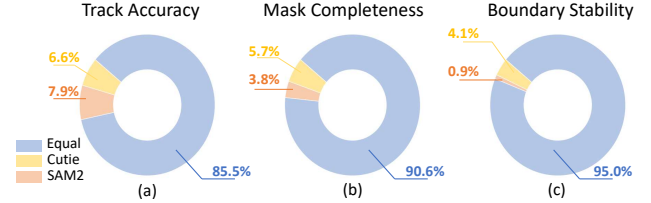


Figure 3. The result of blind review. Through subjective evaluation of three metrics (a) Track Accuracy, (b) Mask Completeness, and (c) Boundary Stability, we found that the Mask results obtained with Cutie-assisted annotation and SAM2-assisted annotation show little difference in performance.

booster factors	M ³ -VOS full			M ³ -VOS core			YouTubeVOS-2019 val				
	\mathcal{T}	\mathcal{J}_{tr}	\mathcal{J}_{cc}	\mathcal{T}	\mathcal{J}_{tr}	\mathcal{J}_{cc}	\mathcal{G}	\mathcal{J}_s	\mathcal{F}_s	\mathcal{J}_u	\mathcal{F}_u
1.01	75.6	66.5	65.2	66.3	55.8	55.5	86.3	85.3	89.7	81.3	88.8
Ours (1.1)	75.6	66.5	65.2	66.3	55.8	55.5	86.8	85.3	89.8	82.1	89.9
1.2	75.8	67.1	65.2	66.4	56.2	55.3	86.7	85.2	89.7	82.1	90.1
1.5	75.8	67.0	65.4	67.4	57.0	56.6	86.5	85.1	89.6	81.7	89.7
2.0	76.0	67.3	66.0	68.0	59.1	57.4	86.6	84.9	89.4	81.9	90.2

Table 4. Comparison with different booster factors.

7. Details of ReVOS Framework

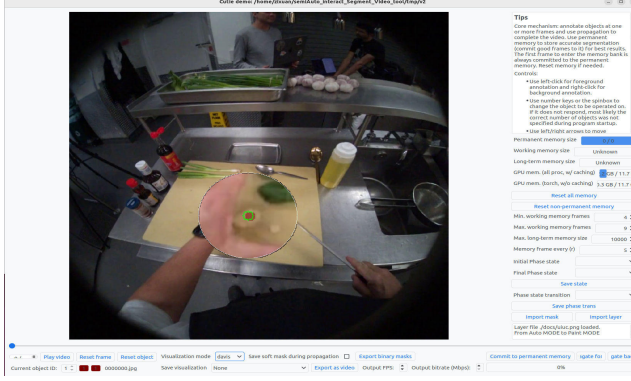
The ReVOS in our implementation adopts Cutie as a backbone model. It is a plug-and-play module and thus can be applied to any mask propagation-based VOS methods. Concerning the implementation of Cutie [1], the details of ReVOS_Cutie are shown in Fig. 5.

8. Comparison with Different Booster Factors

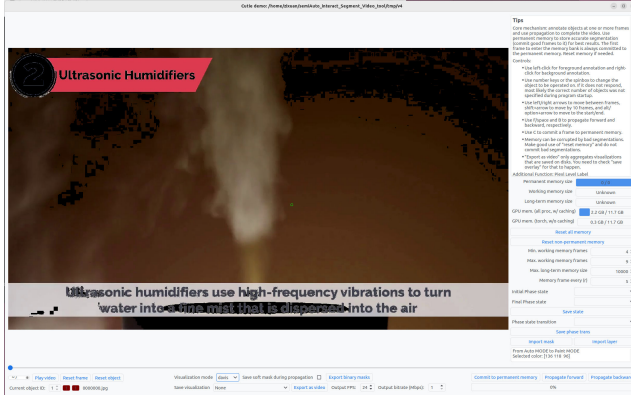
We evaluate different booster factors in Tab. 4.

9. Methods for Object Appearance Changes

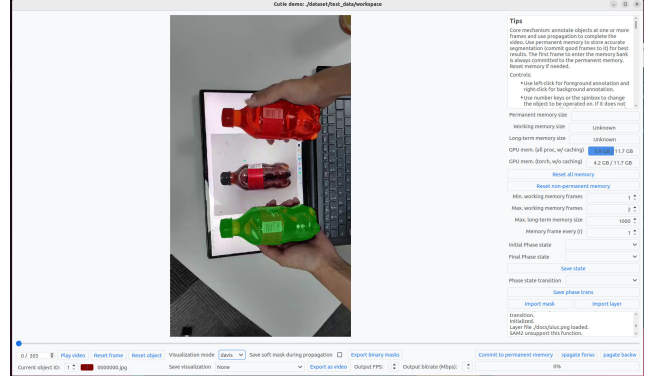
In Tab. 5, we evaluate TAM-VT [2] on M³-VOS, aligning with our setting. Our preliminary results show that our method outperforms others.



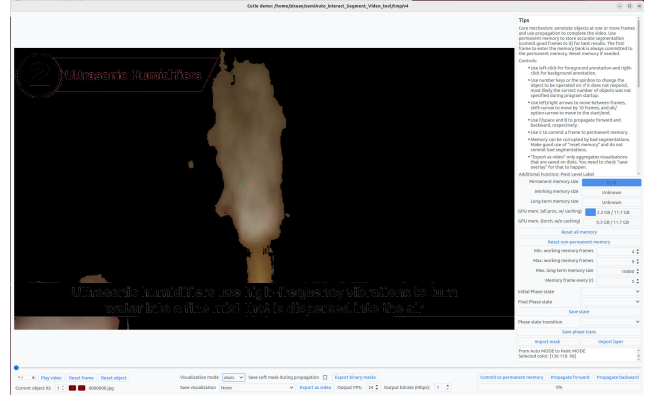
(a) Pixel level: brush with magnifying glass.



(c) Appearance level: select the pixel in the smoke with **high tolerance**.



(b) Object level: apply click prompt to mask object.



(d) Appearance level: select the pixel in the smoke with **low tolerance**.

Figure 4. Multi-level semi-auto annotate tool: Our annotate tool implements three-level annotating (**pixel level**, **appearance level**, **object level**). Using this tool, we can efficiently annotate different objects, such as small objects *garlic* in (a), solid with clear boundaries *cola bottle* in (b) and fluid object *smoke* in (c), (d).

	M ³ -VOS full			M ³ -VOS core			VOST val		
	\mathcal{J}	\mathcal{J}_{tr}	\mathcal{J}_{cc}	\mathcal{J}	\mathcal{J}_{tr}	\mathcal{J}_{cc}	\mathcal{J}	\mathcal{J}_{tr}	\mathcal{J}_{cc}
TAM-VT	73.5	66.1	62.6	58.2	48.9	48.8	40.4	25.3	31.9
<u>Ours</u>	75.6	66.5	65.2	66.3	55.8	55.5	41.0	25.3	31.7

Table 5. Comparison with TAM-VT.

10. Challenge Analysis

In this part, we explore how the size of the object and the velocity of the target object influence the performance of Cutie-ReVOS.

10.1. Definition of Object Size

In our experiment, given a target object o in the image I , its size is measured by the ratio between the mask of the object M_o and the area of the image A_I , according to

$$R(o) = \frac{M_o}{A_I}, \quad (2)$$

where $R(o)$ measures the relative size of the object compared to the Image. M_o is the size of the ground-truth mask of the object O . A_I is the area of Image I .

10.2. Definition of Velocity

Generally, the velocity of an object in an image is defined as the change in the centroid of the bounding box or mask per unit time. However, considering that we cannot measure the relationship between the distance in the image and the actual size of the object, we normalize the velocity based on the size of the object. Given a target object o and the fps f_v of the video clip, the relative velocity or the normalized velocity is defined as follows:

$$v(o) = \frac{D(o)f_v}{M_o}, \quad (3)$$

$$D(o) = c_t(B_o) - c_{t-1}(B_o),$$

where B_o is the bounding box of target object o . $c_t(B)$ is the centroid of the bounding box in the timestamp t . $D(o)$ is the moving distance of the Object in a frame.

10.3. Relation between Challenge and Performance

As the curve in Fig. 6a demonstrated, the smaller the object's area ratio, the more challenging it is for the model to segment. Besides, for small objects, the performance of

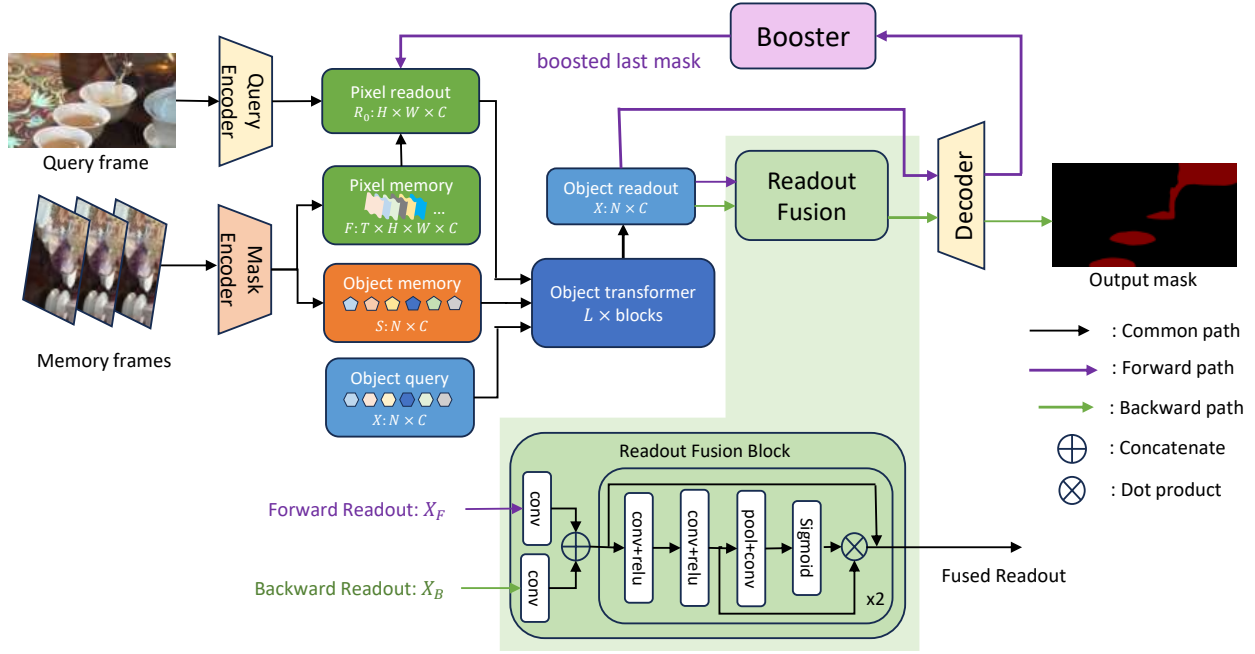


Figure 5. Details of ReVOS (Cutie as backbone) framework. The pixel readout is generated from the fusion of the output of the query encoder, the pixel memory, and the boosted last mask. Then, in the pixel readout, the object memory is queried by the object query in the Object transformer. For each query frame, we calculate its object readout during both its forward (shown in purple axis) and backward propagation process (shown in green axis). During the forward process, the object readout is decoded and boosted in the Booster module. During the backward process, the forward and backward readouts are fused in the Readout Fusion module to create the final output mask.

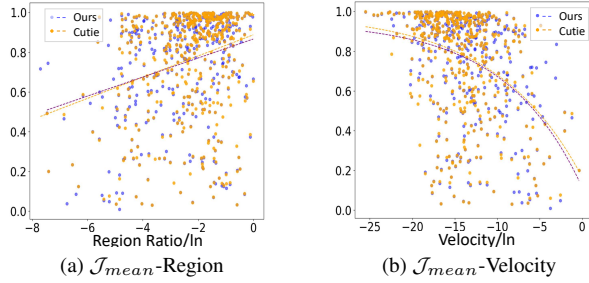


Figure 6. The relation between standard metric Jaccard index \mathcal{J}_{mean} and region size, velocity.

ReVOS-Cutie decreases compared to the original Cutie. For large objects, the situation is reversed.

Similarly, the curve in Fig. 6b indicates that the relative velocity shows a positive correlation with segmentation difficulty. We observe that when the velocity is more extreme, either too slow or too fast, the performance improvement of ReVOS-Cutie becomes more significant.

11. More Failure Cases

In this part, we show more failure cases of the current models in Figs. 7 and 8.

In case 1 (*fry dough*) of Fig. 7, the boiling oil makes it difficult to separate the boundaries of the dough sticks accurately. Even for some models, the boiling oil causes tracking loss. However, our method improves the segment accuracy with visual distribution.

In case 2 (*assemble puzzles*) of Fig. 7, the intra-solid transition with multi-instances usually causes instance confusion because of the similarity distribution. However, our method is more robust when facing the intra-solid transition with multi-instances.

In case 1 (*cut apples*) of Fig. 8, the tracking loss and mask incompleteness usually happen when the white pulp leaks out. The performance in the intra-solid phase transition with the color change challenge is not so good.

In case 2 (*pour tea*) of Fig. 8, the flow of tea liquid into the tea cup is always accompanied by tracking loss. Besides, transparent teapots and tea liquids also suffer from similar interference and are confused. Although our method improves this situation slightly, this multi-challenge case still has improved space.

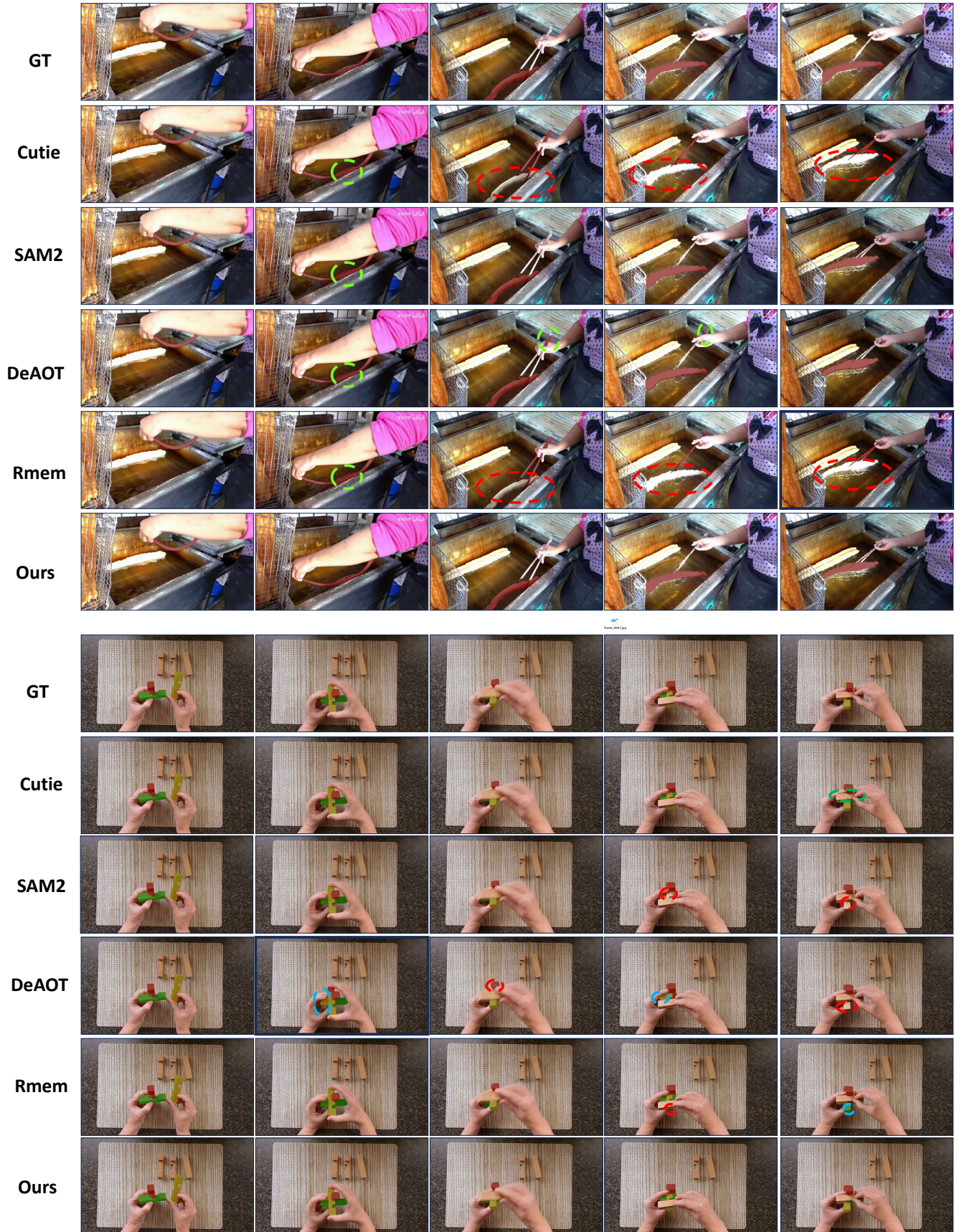


Figure 7. Failure case 3 (*fry dough*) and 4 (*assemble puzzles*) in different models. (Red circle: false-negative region; Green circle: false-positive region; Blue circle: confuse-instance region).

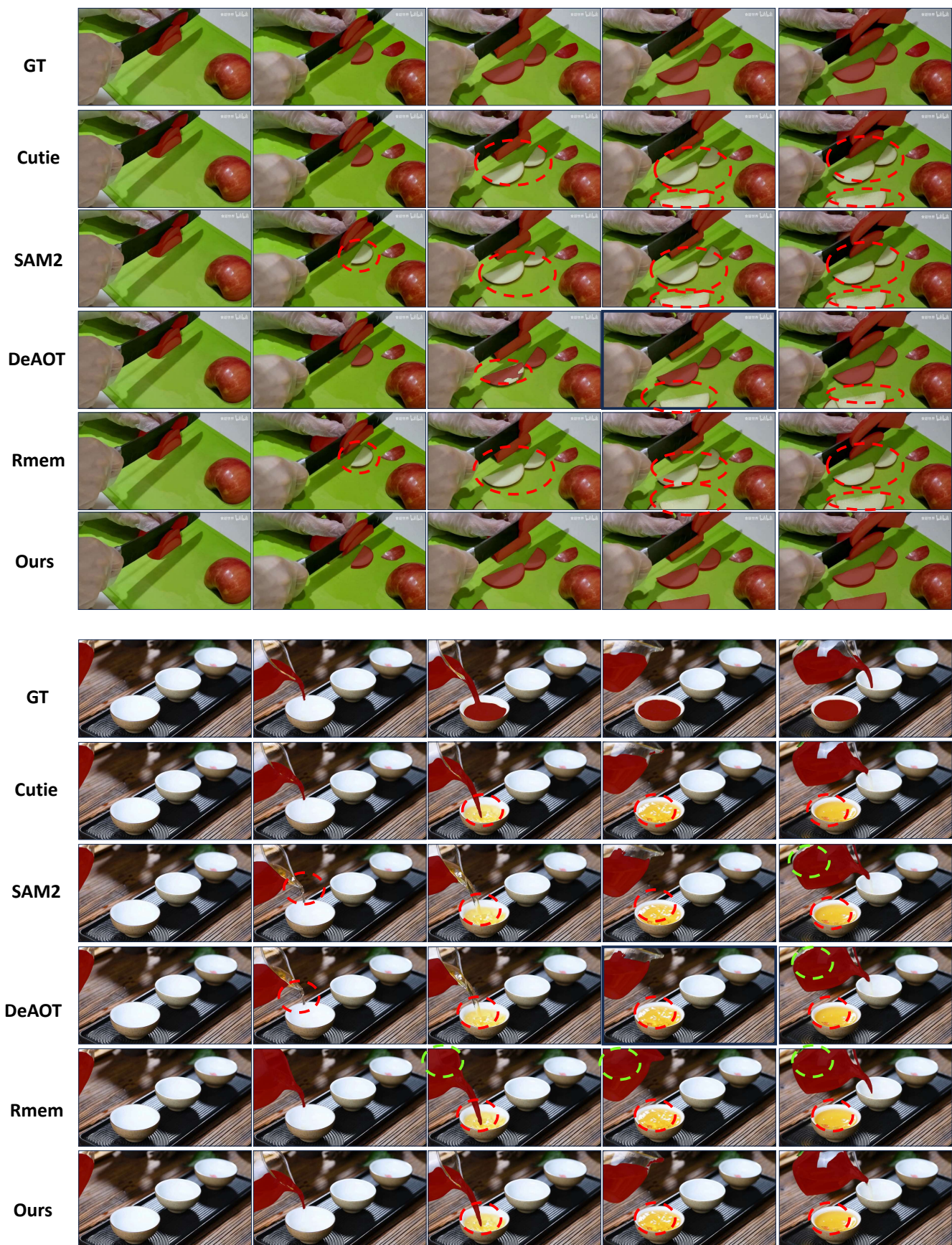


Figure 8. Failure case 1 (*cut apples*) and 2 (*pour tea*) in different models. (Red circle: false-negative region; Green circle: false-positive region).

References

- [1] Ho Kei Cheng, Seoung Wug Oh, Brian Price, Joon-Young Lee, and Alexander Schwing. Putting the object back into video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3151–3161, 2024. [3](#), [4](#)
- [2] Raghav Goyal, Wan-Cyuan Fan, Mennatullah Siam, and Leonid Sigal. Tam-vt: Transformation-aware multi-scale video transformer for segmentation and tracking, 2024. [4](#)
- [3] 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. [2](#)
- [4] Pavel Tokmakov, Jie Li, and Adrien Gaidon. Breaking the” object” in video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22836–22845, 2023. [2](#)
- [5] Jiangwei Yu, Xiang Li, Xinran Zhao, Hongming Zhang, and Yu-Xiong Wang. Video state-changing object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20439–20448, 2023. [1](#), [2](#)