Tracking and Understanding Object Transformations

Yihong Sun, Xinyu Yang, Jennifer J. Sun, Bharath Hariharan

Cornell University

Motivation & Task

Motivation & Task

- Object often undergo **transformations** that can alter their appearance / geometry / identity.
    - Understanding and tracking these transformations is important (e.g., pre- and post-conditions)

Motivation & Task

- Object often undergo **transformations** that can alter their appearance / geometry / identity.
  - Understanding and tracking these transformations is important (e.g., pre- and post-conditions)

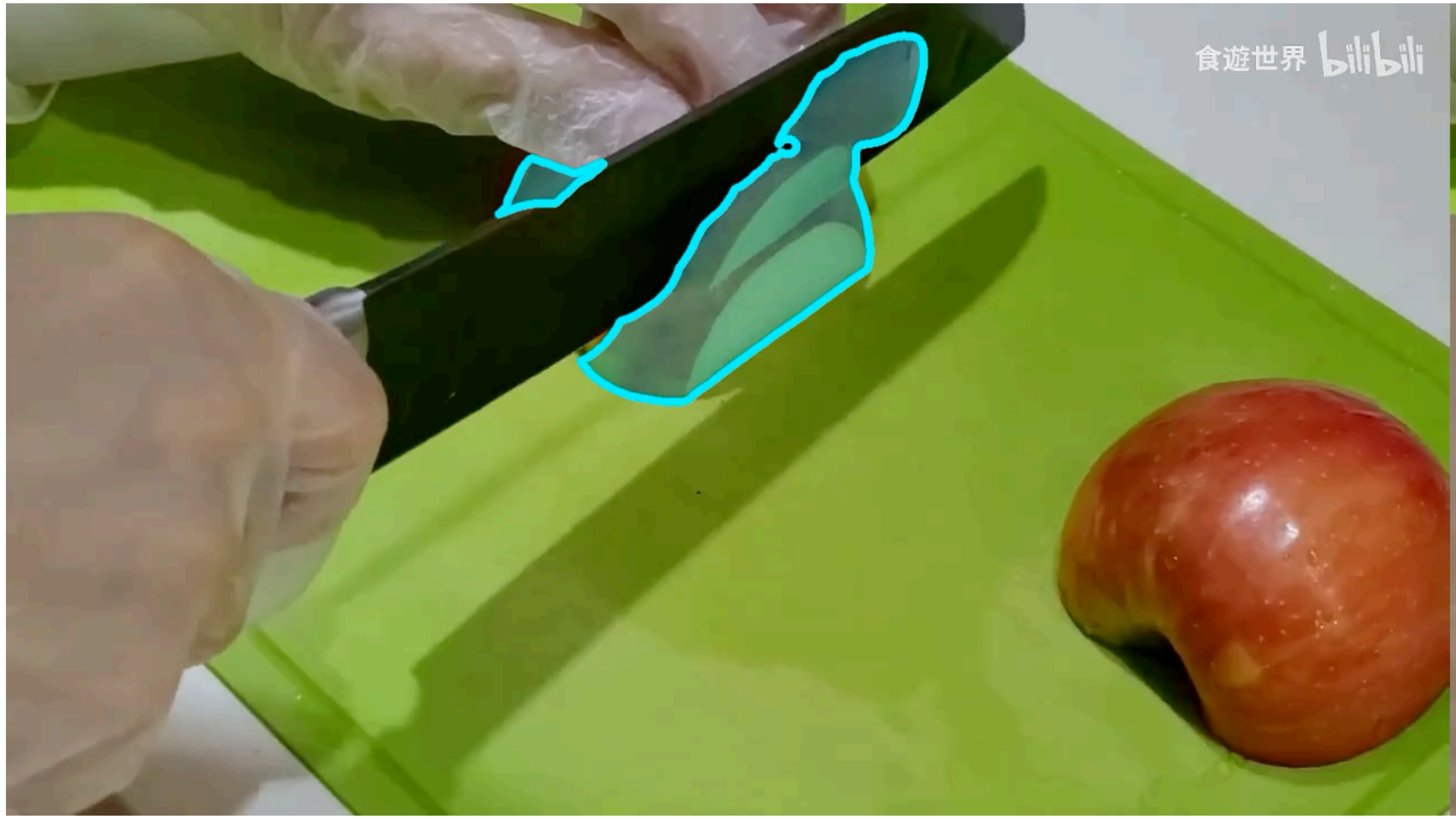
Motivation & Task

- Object often undergo **transformations** that can alter their appearance / geometry / identity.
  - Understanding and tracking these transformations is important (e.g., pre- and post-conditions)
- We propose **Track Any State**
  - When given a video and an object prompt, we map out how the object evolves over time, detect and describe state changes, and track the resulting objects of these changes.

Motivation & Task

- Object often undergo **transformations** that can alter their appearance / geometry / identity.
  - Understanding and tracking these transformations is important (e.g., pre- and post-conditions)
- We propose **Track Any State**
  - When given a video and an object prompt, we map out how the object evolves over time, detect and describe state changes, and track the resulting objects of these changes.

Main challenge

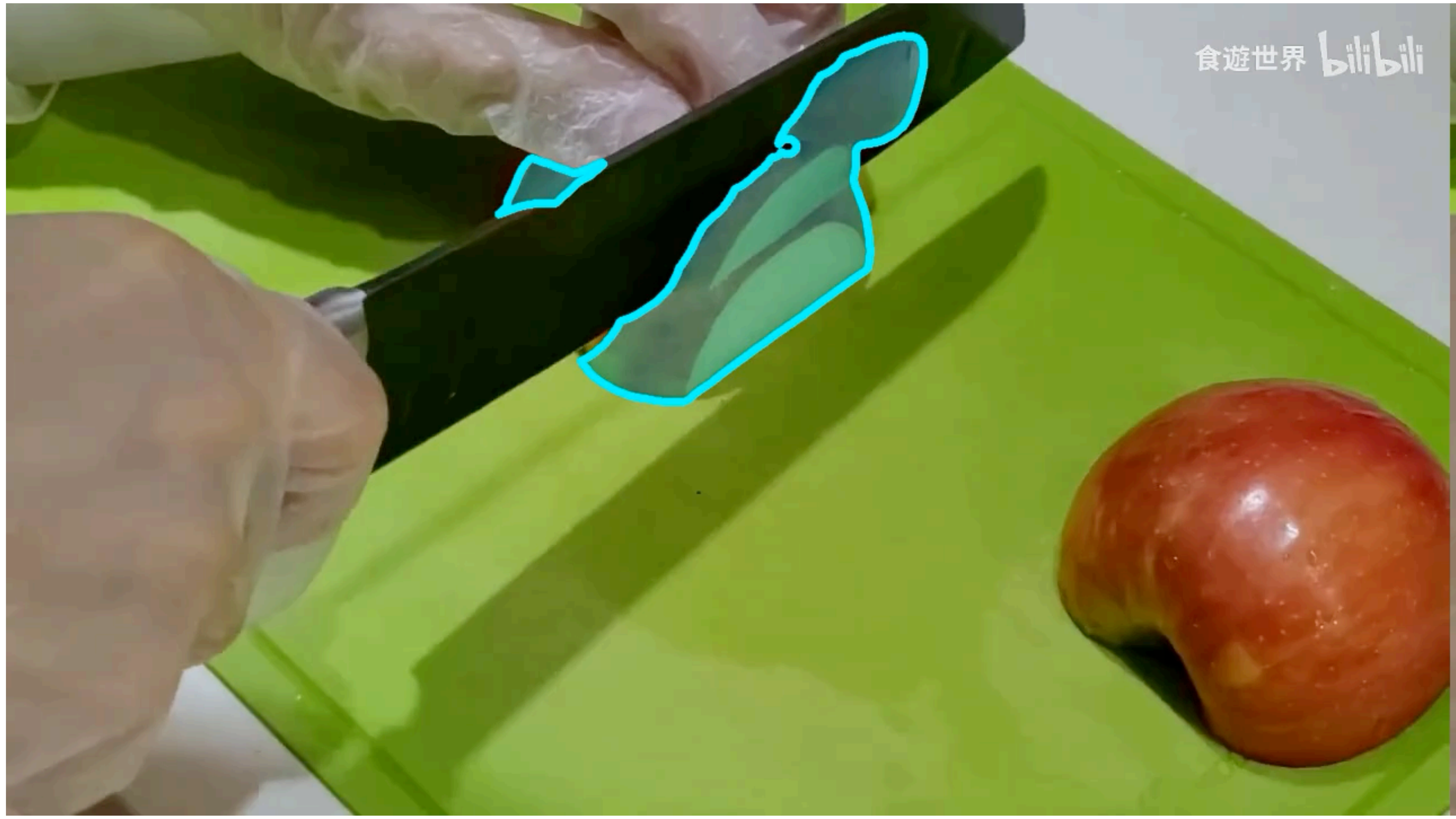


Ground Truth

SAM2.1

Main challenge

- Existing object trackers often fail to keep track of the **complete object** after transformation.

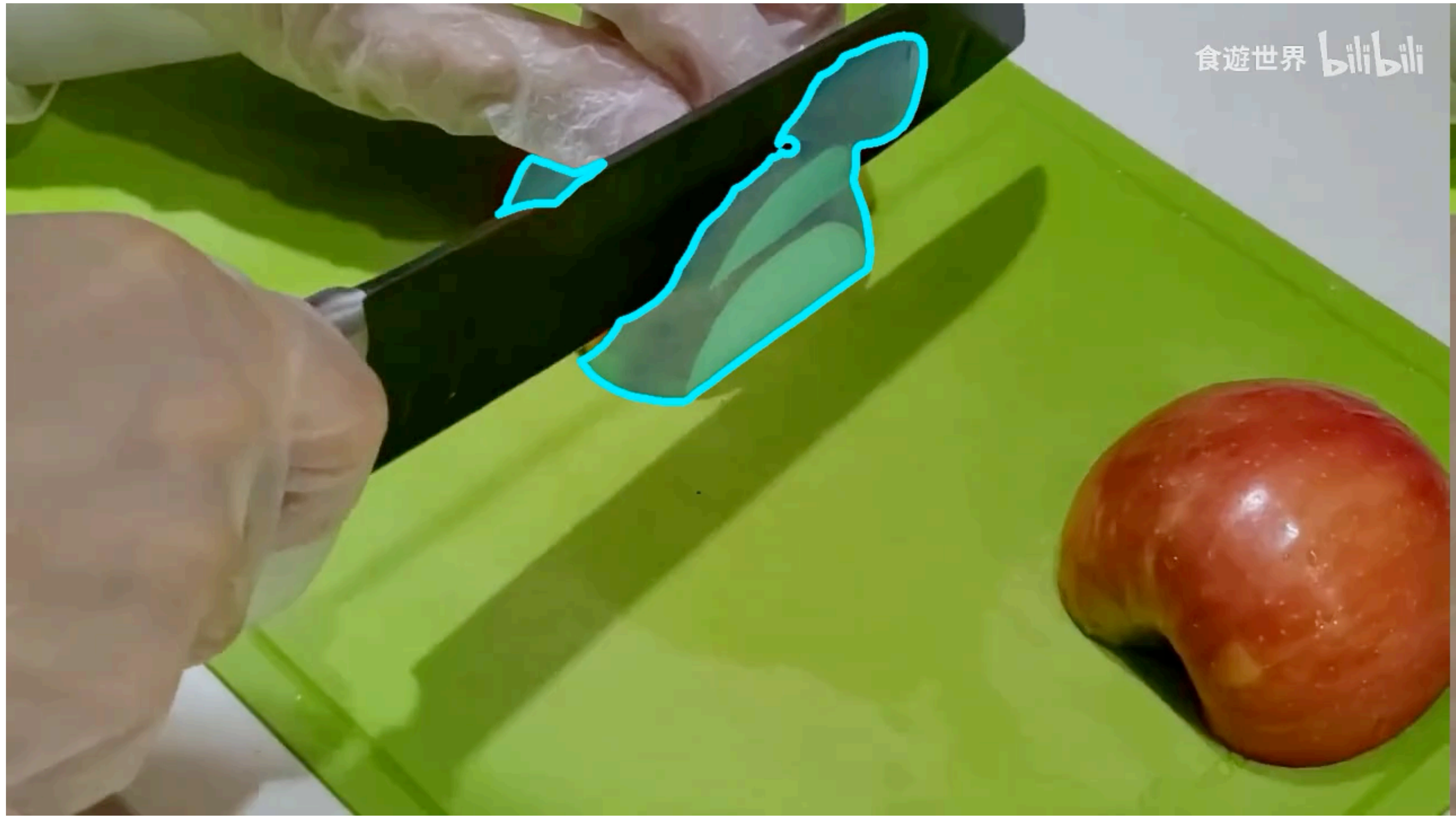


Ground Truth

SAM2.1

Main challenge

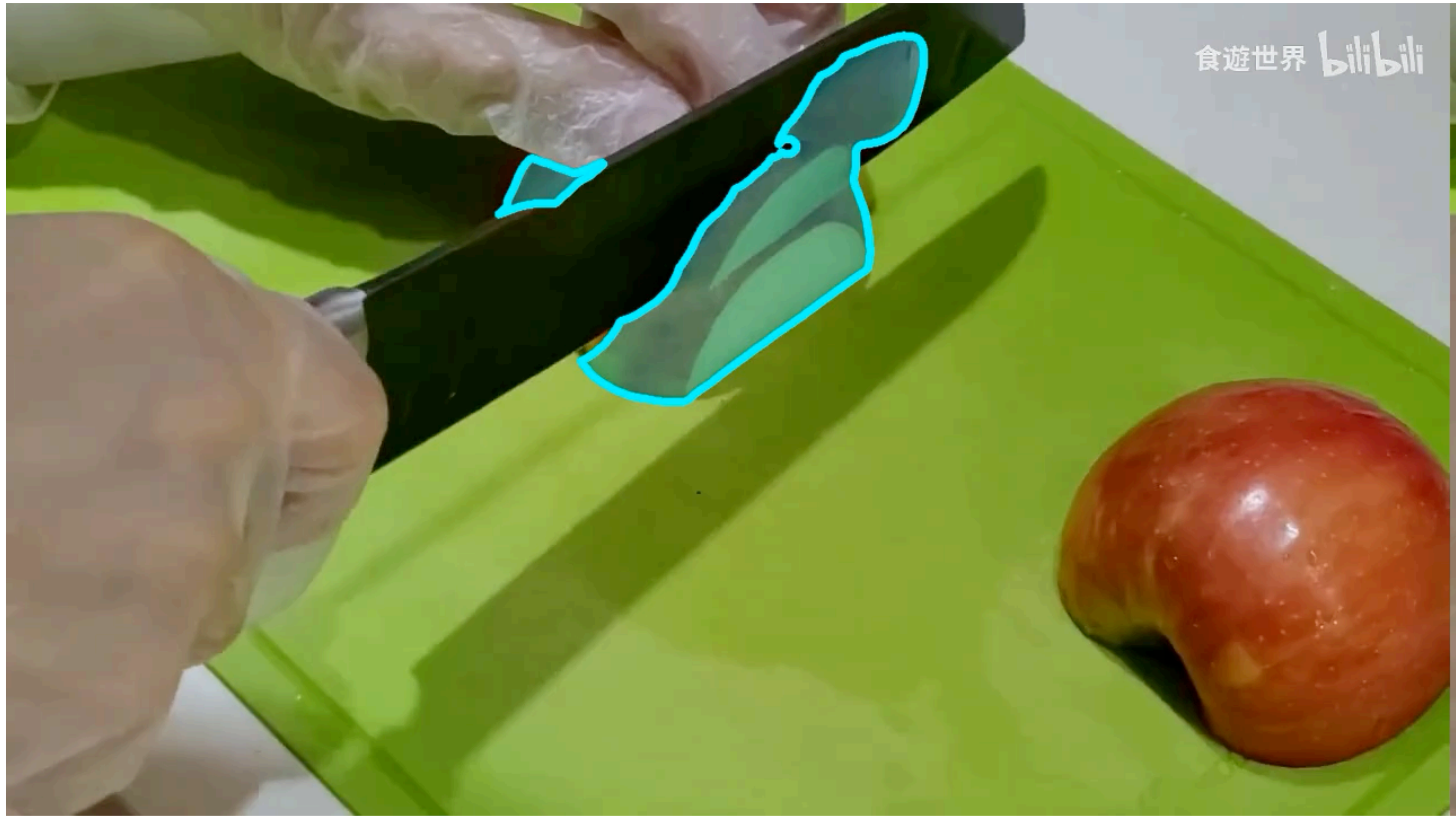
- Existing object trackers often fail to keep track of the **complete object** after transformation.
  - These failures are often caused by object-part separations, appearance changes, shape deformations, etc.



Ground Truth

SAM2.1

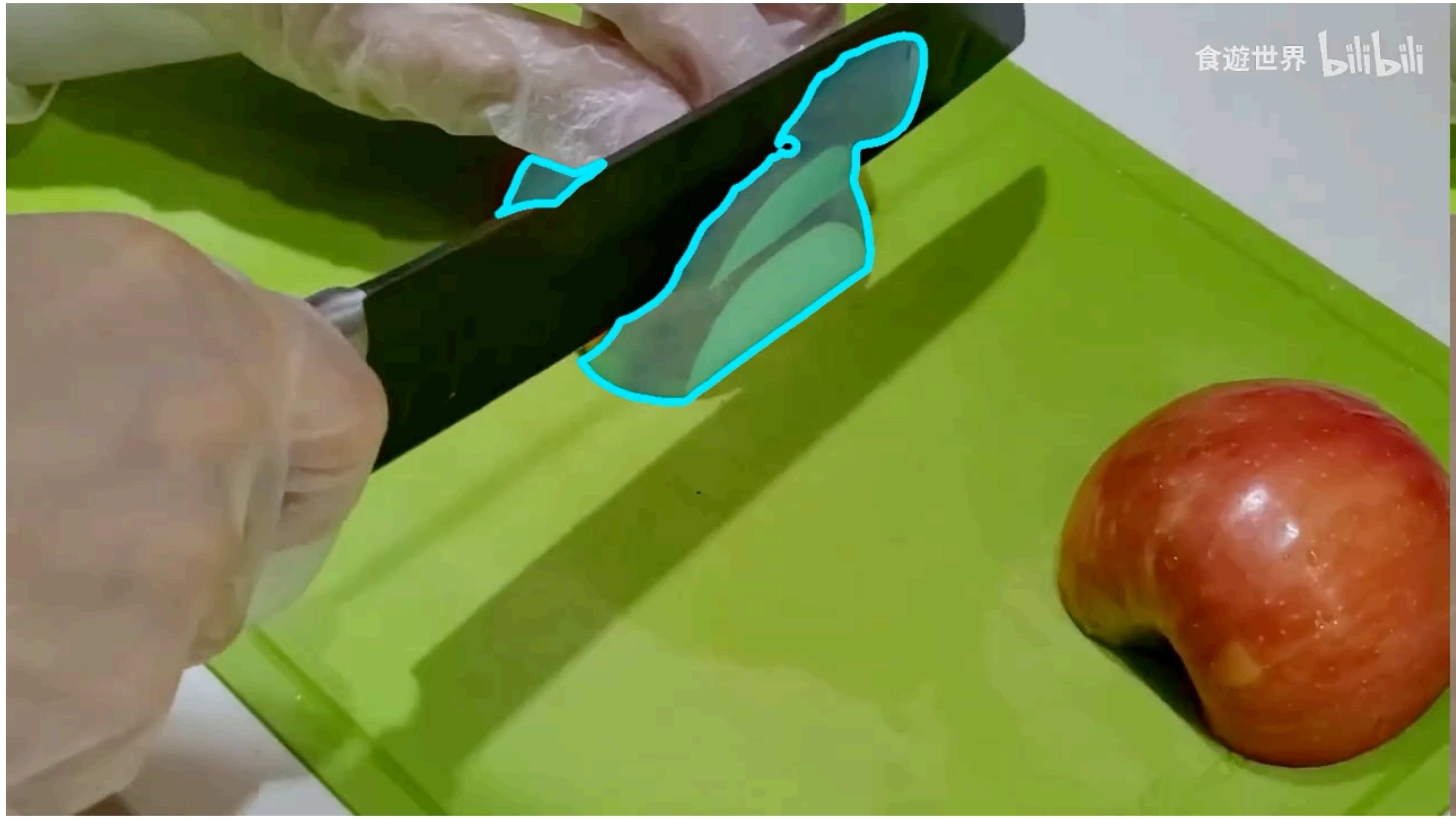
Key Insight



Ground Truth

SAM2.1

Key Insight

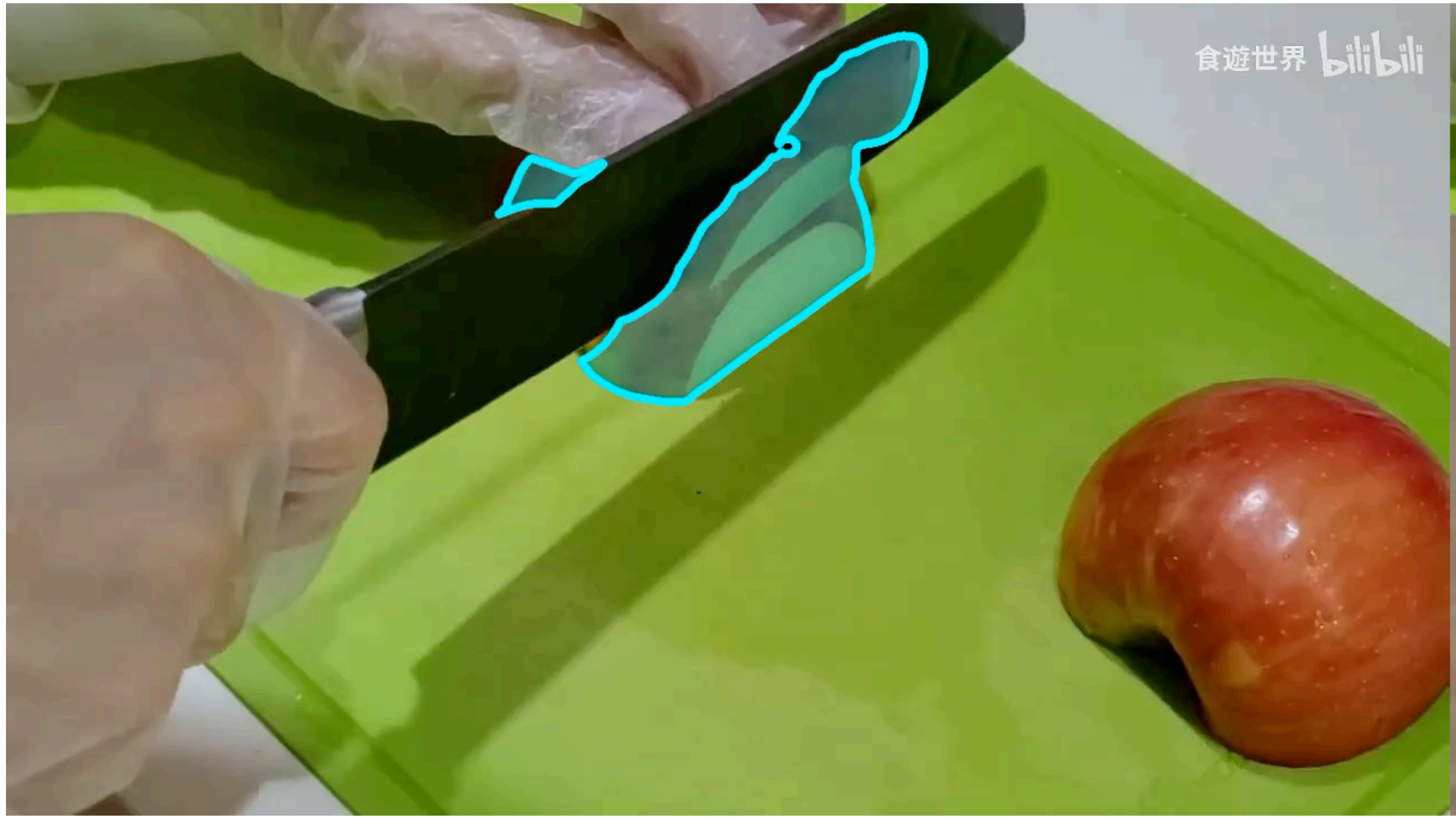


Ground Truth

SAM2.1

Key Insight

- Object tracking errors are **asymmetric**: false negatives >> false positives
  - The missing objects (false negatives) are often caused by appearance-altering **transformations**.

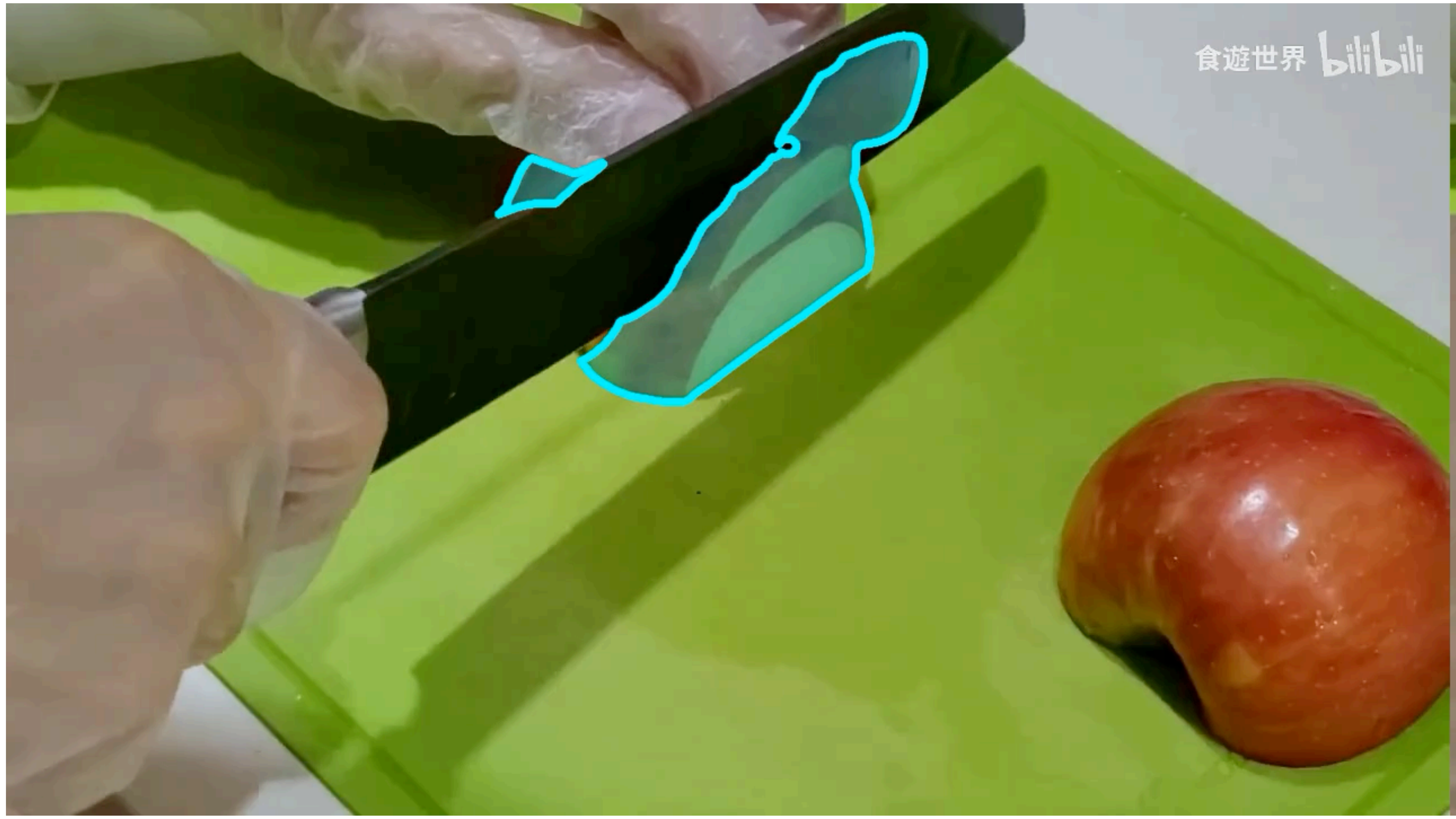


Ground Truth

SAM2.1

Key Insight

- Object tracking errors are **asymmetric**: false negatives >> false positives
  - The missing objects (false negatives) are often caused by appearance-altering **transformations**.
- Recovering them reveals **when and where** these transformations occurred!

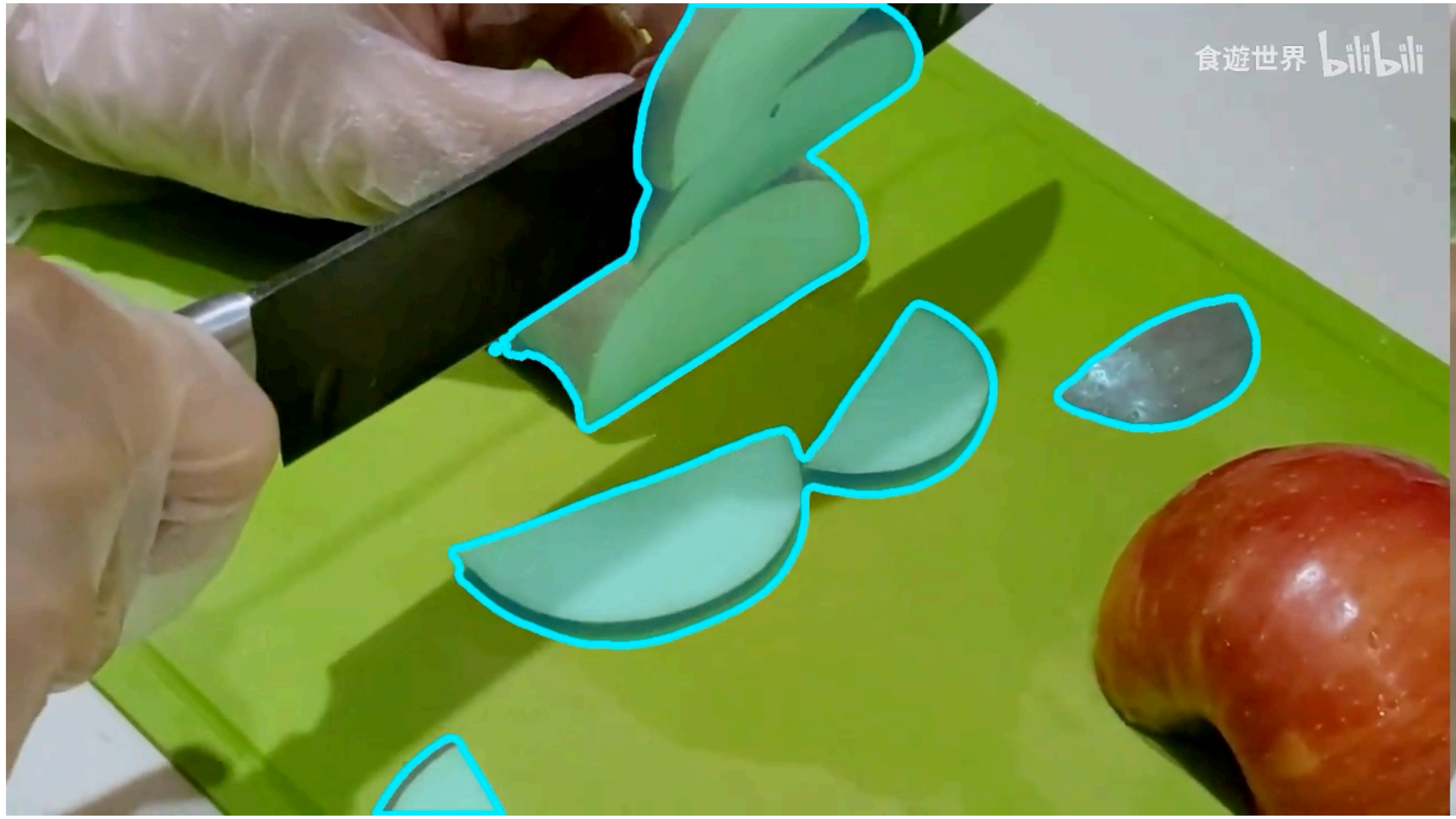


Ground Truth

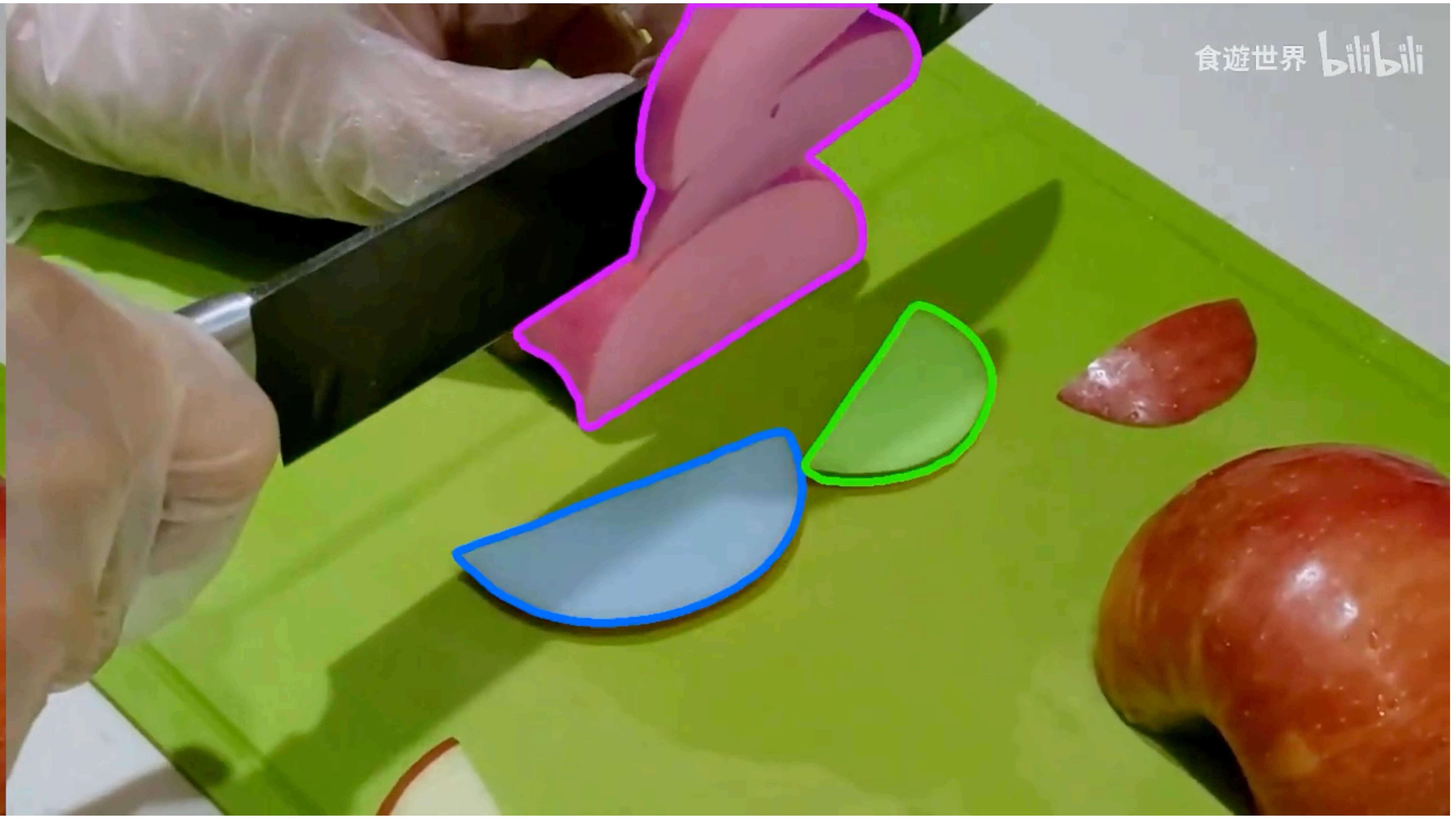
SAM2.1

Our Solution

- We propose **TubeletGraph**, a zero-shot framework that recovers missing objects post-transformation and constructs a state graph to detect and describe the underlying transformations.



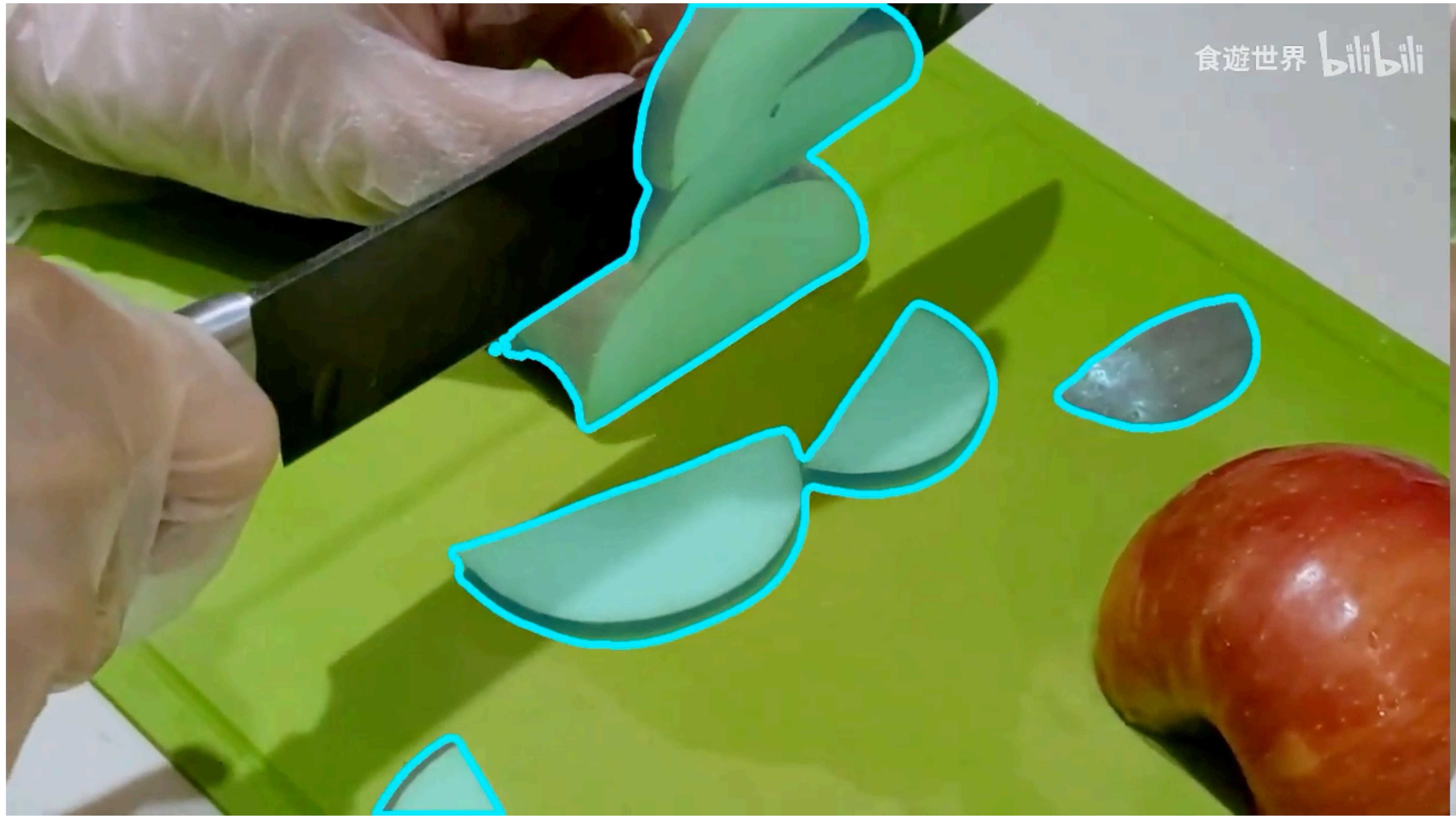
Ground Truth



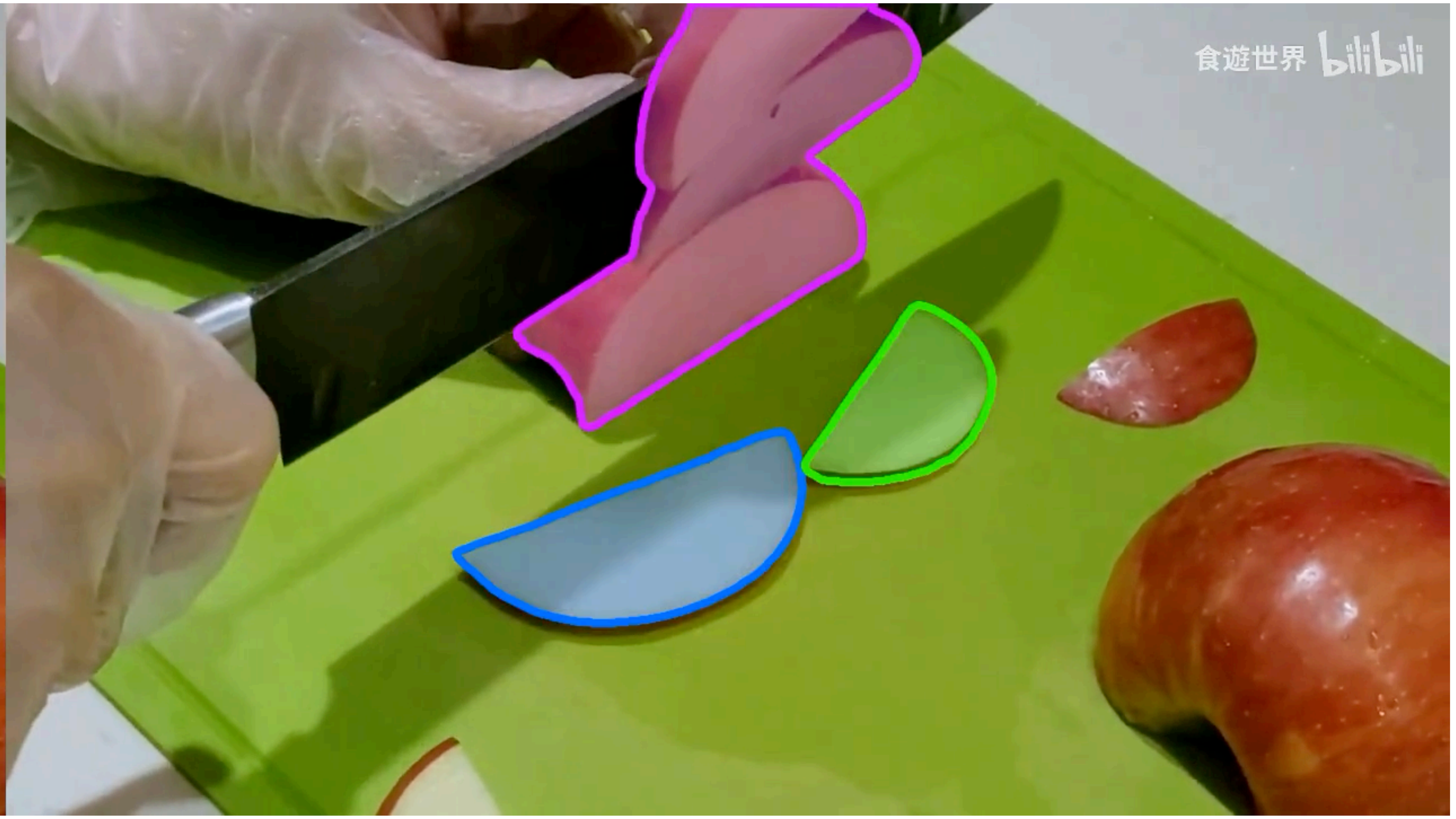
TubeletGraph

Our Solution

- We propose **TubeletGraph**, a zero-shot framework that recovers missing objects post-transformation and constructs a state graph to detect and describe the underlying transformations.



Ground Truth



TubeletGraph

Our Solution

- We propose **TubeletGraph**, a zero-shot framework that recovers missing objects post-transformation and constructs a state graph to detect and describe the underlying transformations.

Our Solution

- We propose **TubeletGraph**, a zero-shot framework that recovers missing objects post-transformation and constructs a state graph to detect and describe the underlying transformations.
  - It constructs a **spatiotemporal partition** of the video by tracking all regions and recovers missing objects via semantic and spatial proximity priors.

Our Solution

- We propose **TubeletGraph**, a zero-shot framework that recovers missing objects post-transformation and constructs a state graph to detect and describe the underlying transformations.
    - It constructs a **spatiotemporal partition** of the video by tracking all regions and recovers missing objects via semantic and spatial proximity priors.
    - Recovered objects serve as **transformation markers**: It then prompts VLMs for descriptions

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
   - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
   - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

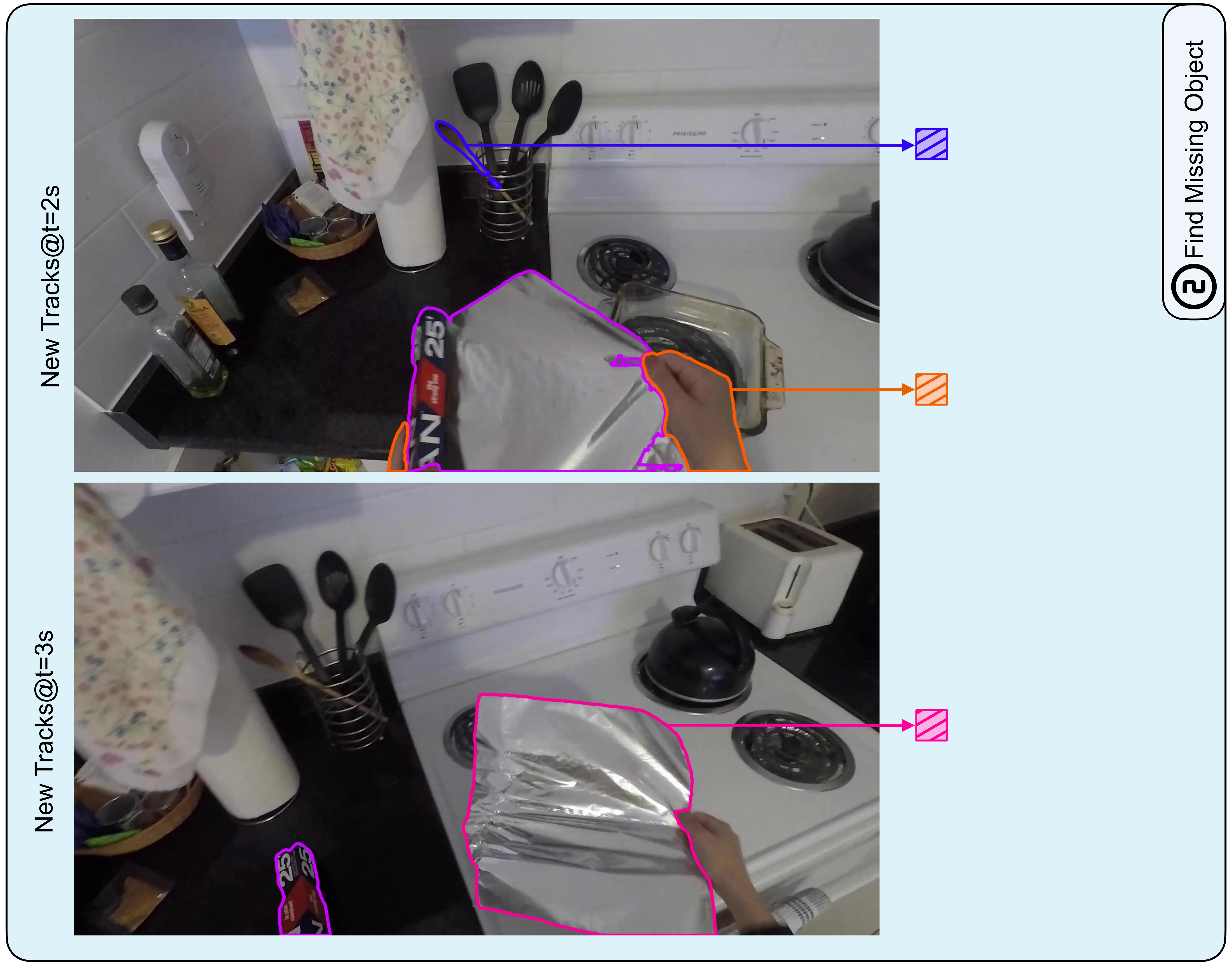
1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
   - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

Method Overview

1. We first obtain a spatiotemporal partition of the input video with provided object prompt.
    - Track all regions from the first frame and initiate new tubelets when untracked pixels emerge.

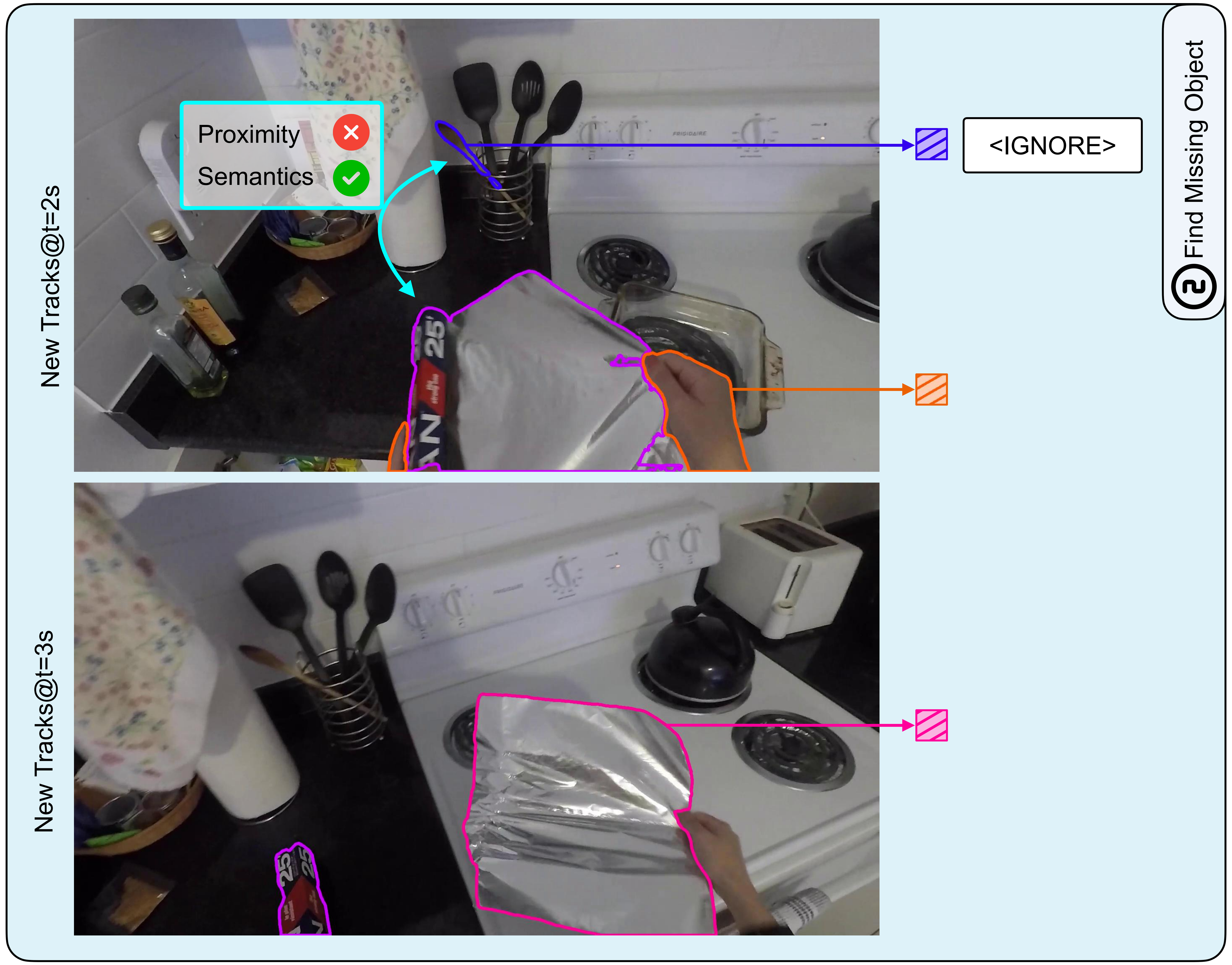
Method Overview

2. For each newly-emergent entity region, we reason about its proximity and semantic consistency and only keep the candidates that satisfy both.



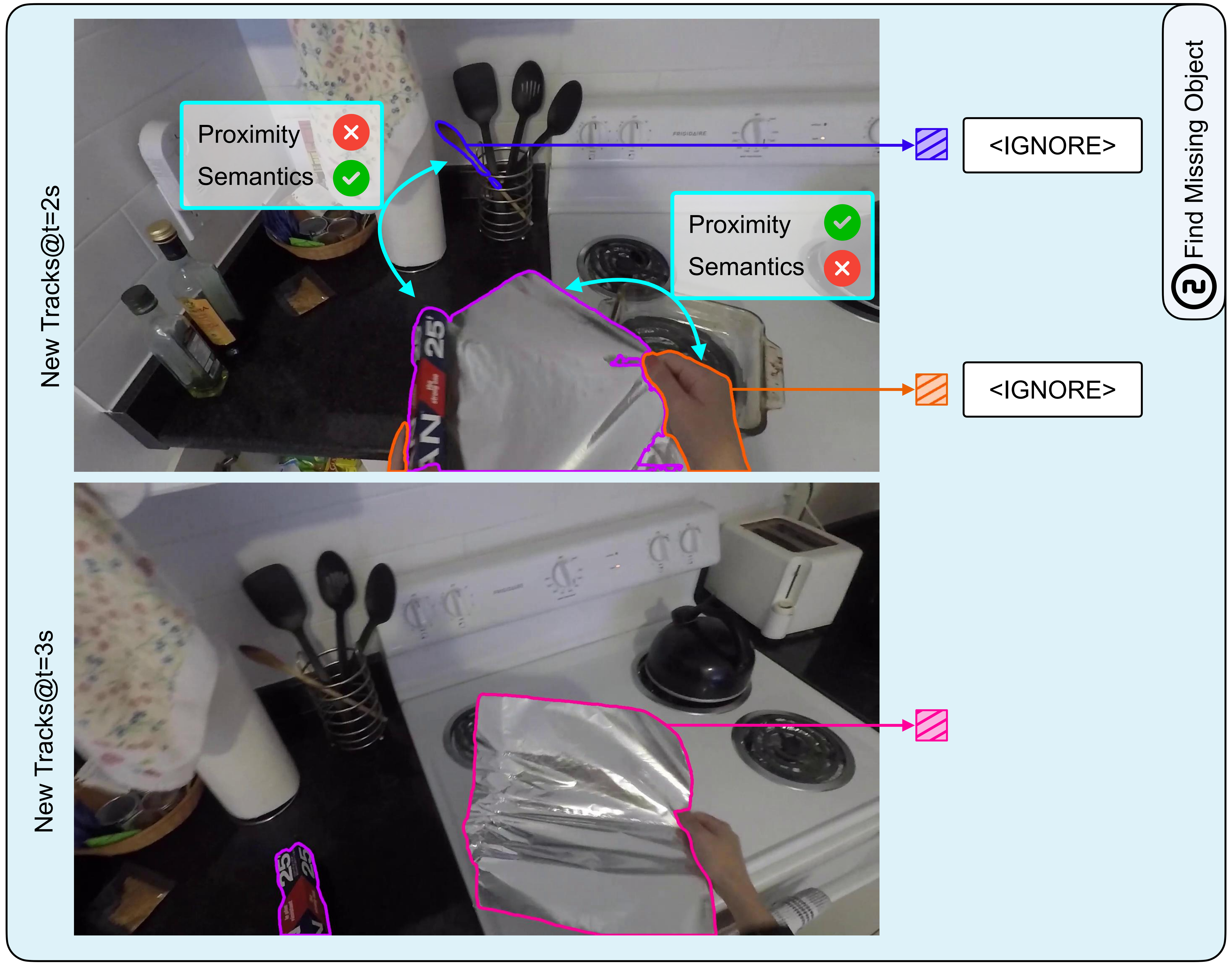
Method Overview

2. For each newly-emergent entity region, we reason about its proximity and semantic consistency and only keep the candidates that satisfy both.



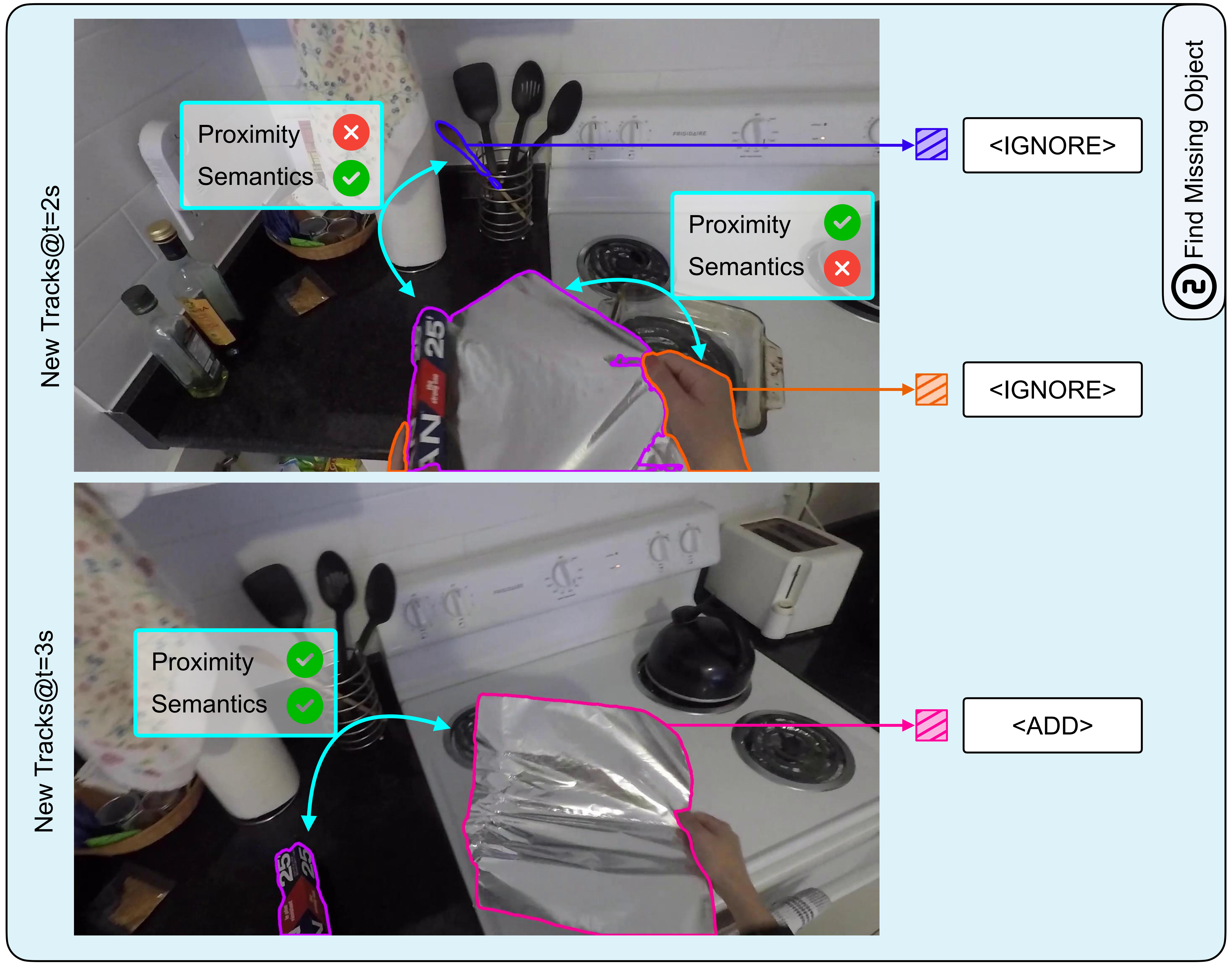
Method Overview

2. For each newly-emergent entity region, we reason about its proximity and semantic consistency and only keep the candidates that satisfy both.



Method Overview

2. For each newly-emergent entity region, we reason about its proximity and semantic consistency and only keep the candidates that satisfy both.



Method Overview

3. For each recovered region, we prompt VLM to describe the transformation and resulting objects.

Method Overview

3. For each recovered region, we prompt VLM to describe the transformation and resulting objects.

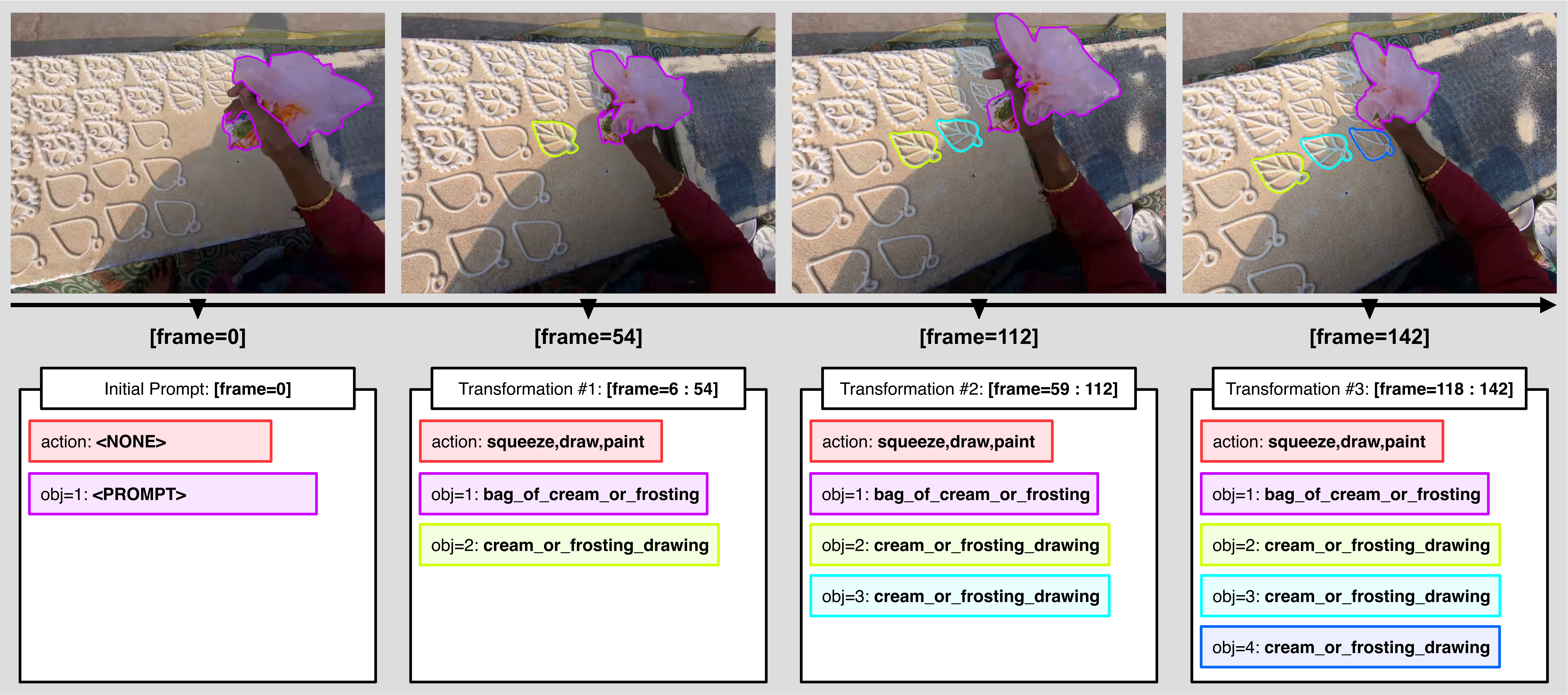
Method Overview

3. For each recovered region, we prompt VLM to describe the transformation and resulting objects.

Method Overview

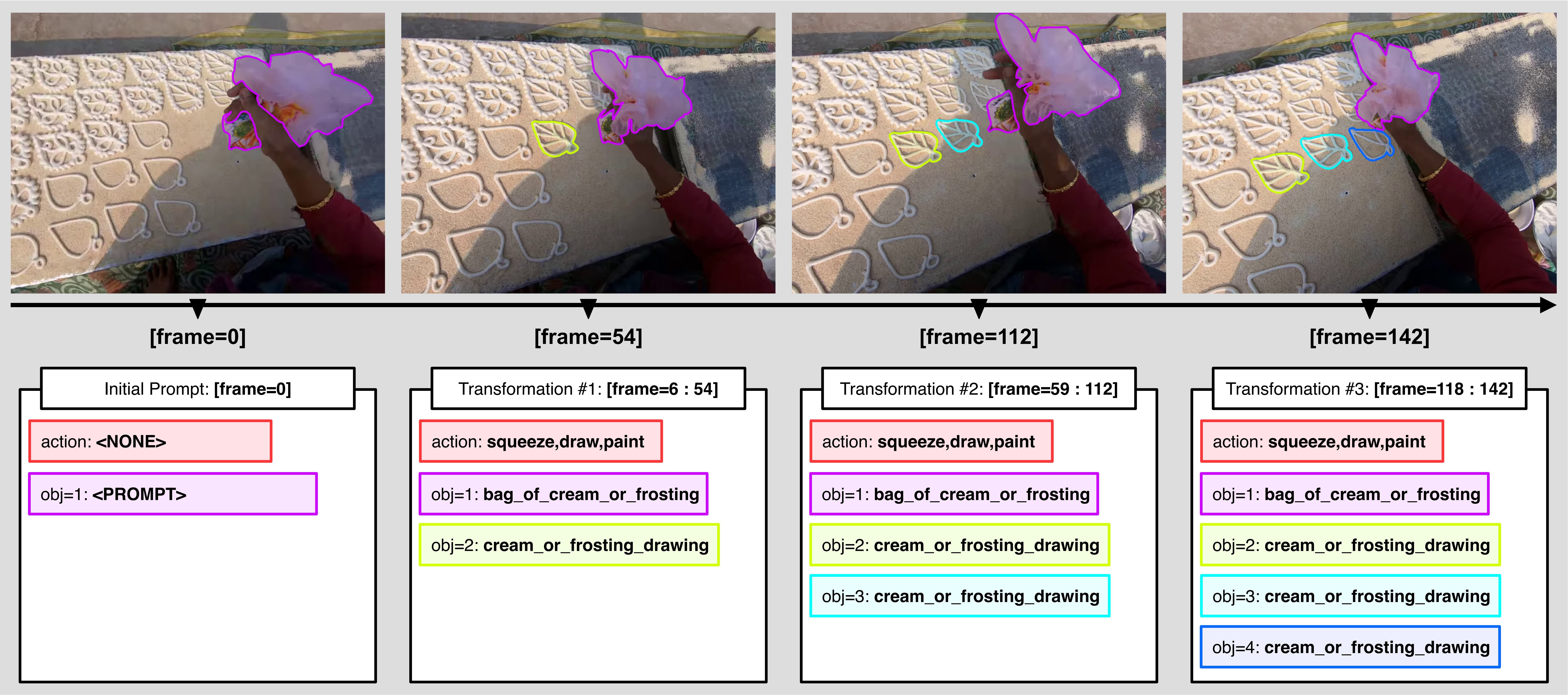
4. Finally, we obtain the complete object tracks along with the predicted state graph.

VOST-TAS


[frame=0]
[frame=54]
[frame=112]
[frame=142]
Initial Prompt: [frame=0]
action: <NONE>
obj=1: <PROMPT>
Transformation #1: [frame=6 : 54]
action: squeeze,draw,paint
obj=1: bag_of_cream_or_frosting
obj=2: cream_or_frosting_drawing
Transformation #2: [frame=59 : 112]
action: squeeze,draw,paint
obj=1: bag_of_cream_or_frosting
obj=2: cream_or_frosting_drawing
obj=3: cream_or_frosting_drawing
Transformation #3: [frame=118 : 142]
action: squeeze,draw,paint
obj=1: bag_of_cream_or_frosting
obj=2: cream_or_frosting_drawing
obj=3: cream_or_frosting_drawing
obj=4: cream_or_frosting_drawing

VOST-TAS

- Finally, we introduce VOST-TAS (TrackAnyState), an extended version of the VOST validation set with explicit transformation annotations for tracking and understanding object state changes in videos.
    - It contains 57 video instances, 108 transformations, and 293 annotated resulting objects.



Results

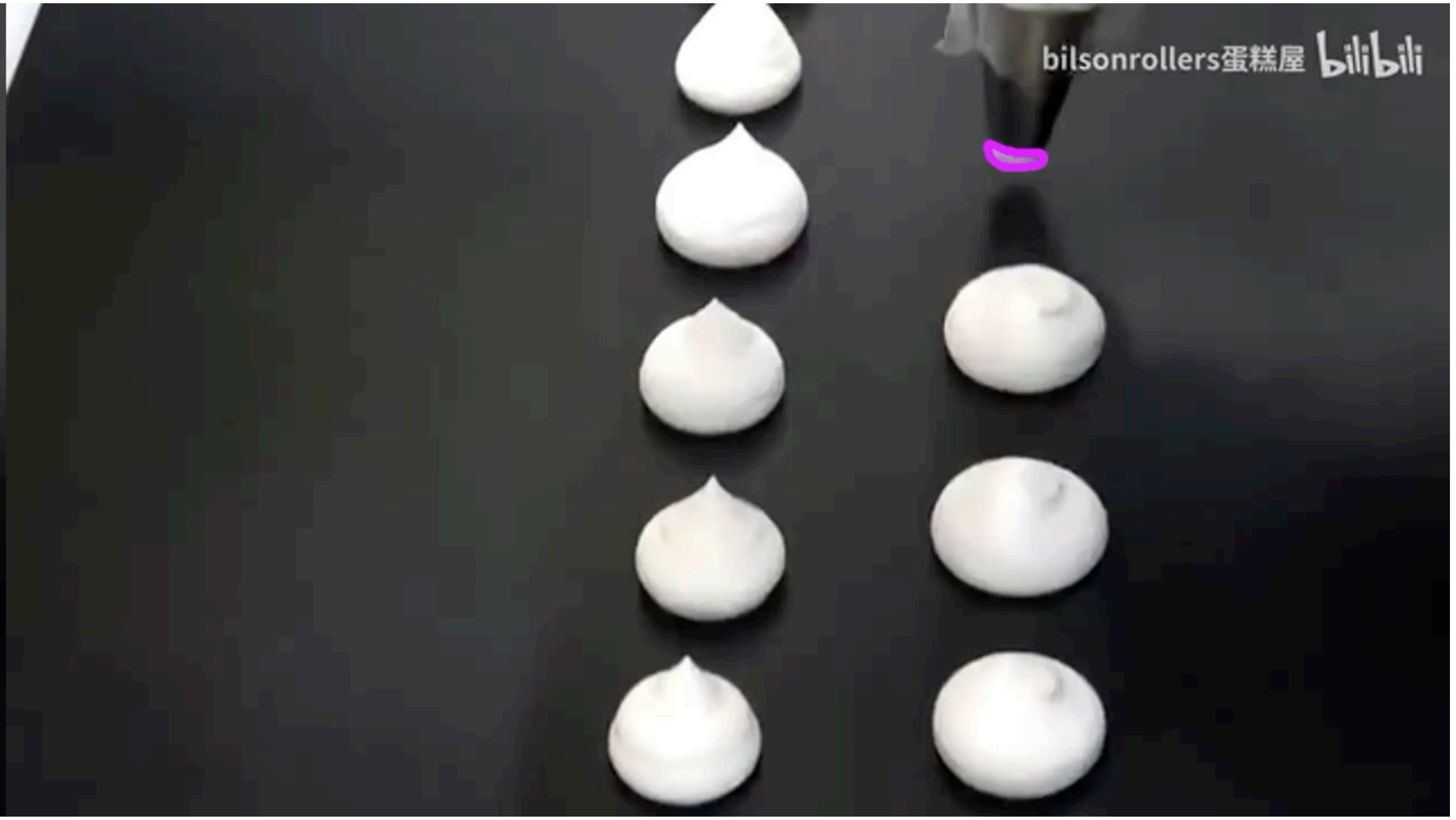
- **State-of-the-art tracking:** TubeletGraph achieves superior performance on transformation datasets (e.g., VOST, VSCOS).
- **Novel state graph capabilities:** It demonstrates promising capabilities in state graph prediction, as evaluated on VOST-TAS, with strong semantic accuracy and temporal precision, though overall transformation recall remains challenging.

Method	Tracking						State Graph					
	VOST		VSCOS		M^3-VOS		Sem. Acc.		Temp. Loc.		Overall	
	$\mathcal{J}$	$\mathcal{J}_{tr}$	$\mathcal{J}$	$\mathcal{J}_{tr}$	$\mathcal{J}$	$\mathcal{J}_{tr}$	$\mathcal{S}_V$	$\mathcal{S}_O$	$\mathcal{T}_P$	$\mathcal{T}_R$	$\mathcal{H}_{ST}$	$\mathcal{H}$
SAM2	46.1	29.4	72.5	67.1	71.3	59.8	-	-	-	-	-	-
SAM2Long	46.4	29.1	<u>73.0</u>	<u>68.6</u>	70.2	58.7	-	-	-	-	-	-
SAM2.1	48.4	32.4	72.0	66.9	71.3	59.3	-	-	-	-	-	-
DAM4SAM	48.8	33.6	71.3	66.0	72.2	61.3	-	-	-	-	-	-
SAMURAI	<u>49.8</u>	<u>34.0</u>	71.8	66.9	<u>72.6</u>	<u>61.6</u>	-	-	-	-	-	-
Ours	**51.0**	**36.9**	**75.9**	**72.2**	**74.2**	**64.4**	81.8	72.3	**43.1**	**20.4**	**12.0**	**6.5**

Results

Input Video + Prompt Object

Predicted Object Tracks



Piping bag tip

Predicted State Graph

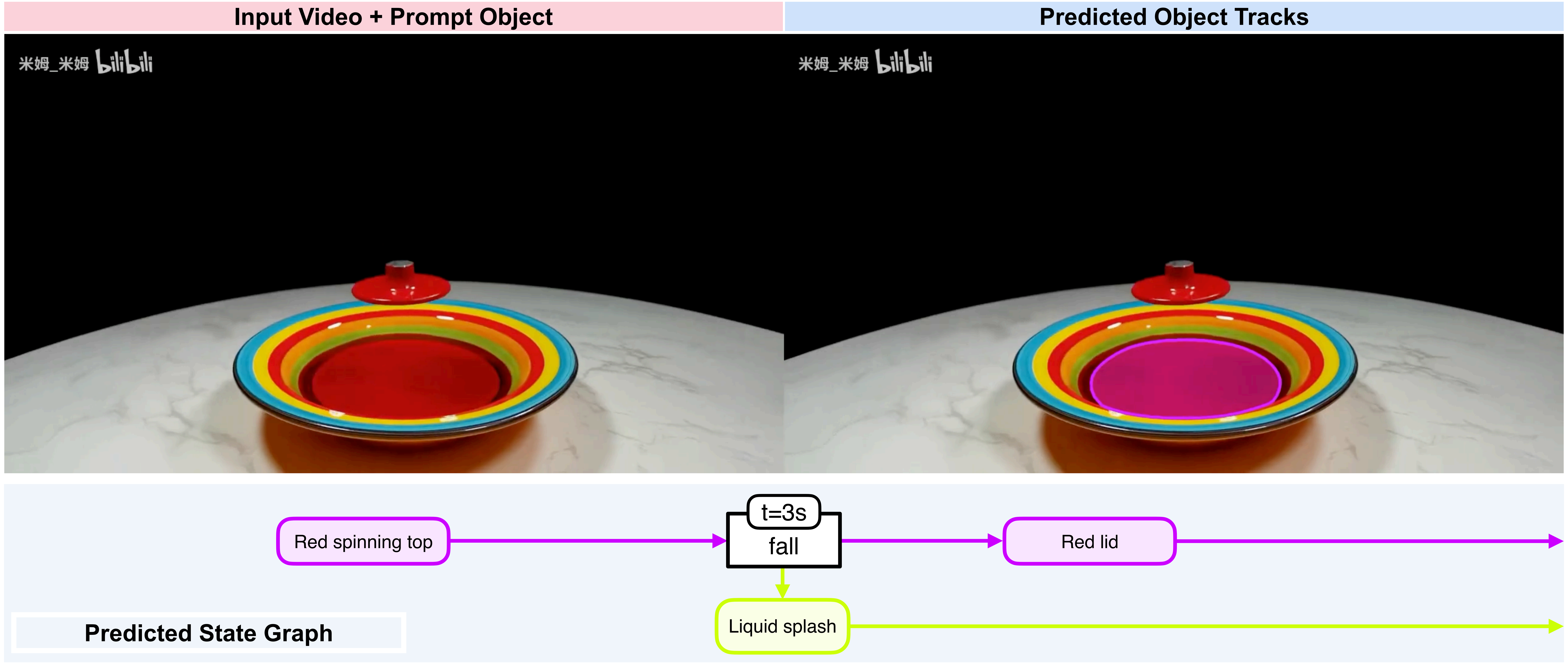
Results

Input Video + Prompt Object
Predicted Object Tracks
bilsonrollers蛋糕屋 bilibili
bilsonrollers蛋糕屋 bilibili
Piping bag tip
t=1s dispense
t=2s dispense
t=4s dispense
t=6s dispense
t=6s dispense
t=6s dispense
t=7s dispense
t=8s dispense
t=8s dispense
Piping tip metal
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
meringue dollop white
Predicted State Graph

Thank you!

For more results, data and code, please visit

tubelet-graph.github.io



Thank you!

For more results, data and code, please visit
tubelet-graph.github.io