MOSCATO: Predicting Multiple Object State Change through Actions

Supplementary Material

1. Dataset

Comparison. In Table 3, we include a more comprehensive version of dataset statistics. We compare number of object, states, videos and hours in addition to the number of object state changes per object and number of objects per video. We observe that while previous datasets have higher number of hours in duration, the average video duration in the MOSCATO dataset is higher than previous datasets. Therefore, in addition to having fine-grained annotations over objects and states, the longer videos compared to previous datasets makes MOSCATO more appropriate for evaluation of the multiple object state prediction methods.
Annotation. During annotation, we ask annotators to first go through the video to familiarize themselves with the video and its sequence of actions. We then ask them to identify objects involved in actions. It is possible that there instances of object visible during video which are not used at any point in the video. In such case, the annotators are instructed to annotate only the objects that have been involved in at least one action at some point in the video. In the next step, we ask annotators to go through the provided list of states for each object and proceed to annotate states.

2. Pseudo Labeling Annotation

Object State Refinement We provide more qualitative results regarding the pseudo-labeling process. In Fig. 2, we include the set of object states before and after the *Object State Refinement* stage. We observe after this stage LLM recovers from missing states and generates a more comprehensive view of the state of the objects.

LLM Prompts. Here, we include the prompts used for Stage 1(Object State Description), Stage 2(State Description Summarization), and Stage 3(Object State Refinement) in the pseudo-labeling process.

Prompt for Object State Description Stage

A task involves several actions. In each action, a set of objects are involved and these objects may undergo state changes. You are given a sequence of actions.
For each action in the sequence, first list the set of all the objects that would be involved in the action in detail.
Second, trace the state of objects involved in the action using the previous state descriptions provided and give descriptions of attributes for each of the involved objects after the action is performed in detail.
Make sure you stick to all the following rules:
- Make sure you have included all the objects that are a part of the action and all the possible tools and utensils.
- If you are sure that no state change happen for one object, for the state description write ¡None¿.
- Separate different parts of your answer using new line.
- remember that descriptions for the object states should be in detail and attributes that they have after the action is completed.
Format your response same as the following example:
User: Slice the bread, spread butter on the bread, spread jam on the bread.
Assistant:
Slice the Bread:
Objects: Bread, Knife
Bread: The bread is now sliced into pieces.
Knife: None.
Spread Butter on the Bread:
Objects: Bread, Butter, Knife
Bread: The surface of the bread has a thin layer of butter spread over it.
Butter: The butter container has a portion scooped out.
Knife: The knife has some butter on its blade.

Prompt for State Description Summarization Stage

The sentence {Description} is describing the state of the object {Object} after the the action {Action} is completed. If possible give one adjective that best summarize the sentence. If that is not possible, use a two-word phrase to summarize the sentence. If none is possible just answer with none. Just answer with the target word or phrase. If there is a need to use more than one adjectives, separate them by a comma. Make sure you give all the states that can be concluded from the sentence. Some examples are as below:
"User: action: Slice the bread, sentence: The bread is now sliced into pieces, object: bread.
Assistant: Sliced.
User: action: spread butter on the bread, sentence: The surface of the bread pieces has a thin layer of butter spread over it., object: butter.
Assistant: on bread, spread.
User: action: spread jam on the bread, sentence: The

knife has jam residue on it, object: knife.
Assistant: dirty".

We breakdown object state refinement into two steps: 1. Ask LLM if any object state needs updating based on the action(refine). 2. For each action and object, check if any of the states before that action still apply to the object(track). In this way, we make sure the LLM does not change state of an object if an action should not modify that state.

Prompt for Object State Refinement

We want to get the comprehensive set of states for each object in the list as a set of actions happen. We have an initial set but want to check if it needs any updates. You are given the states of all objects involved in the task after the action {Action}. Given the current state, for object {Object}, see if any of the states given for it can be added to the current descriptions to describe the object accurately given the information you have.
Set of possible states for object {Object}: {State set}
Given states of all objects after the action {Act}: {Current state set} answer only with updated states.

Prompt for Object State Tracking

We want to get the comprehensive set of states for each object in the list as a set of actions happen. We have the states from before and after the action, but want to check if there are any state in the before state set that can be added to the after state set for each object. You are given the states of all objects involved in the task before and after the action {Action}. For object {Object}, see if any of the states given in the before action states for {Object} is still accurate to describe {Object} accurately after the action. Make sure the states you are selecting from before state to add to after state does not contradict with the states already in the after action states.
Given states of all objects before the action {Action}: {State set before action} Given states of all objects after the action {Action}: {State set after action} In the first line state your reasoning for keeping the states from before the action in the after the action. In the second line, Answer only with updated states for object {Object}.

3. Experiments

Due to space limitations in the main paper, we include some of the experiments in this section.

TAS Results for EGTEA. TAS(Temporal Action Segmentation) results for the EGTEA subset of MOSCATO is included in Tab. 1.

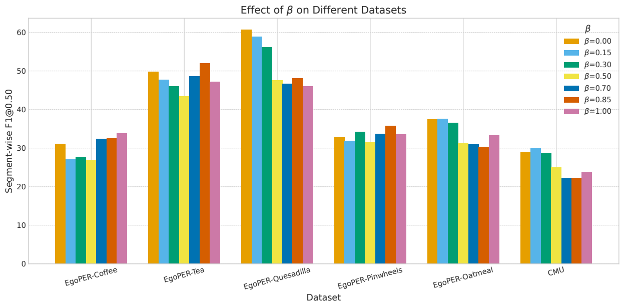


Figure 1. Ablation Study on the effect of the hyperparameter β on the segment-wise F1@0.5.

Ablation on β. We use the hyperparameter β in order to determine the weighting of each MOSP and SAI branches. We perform experiments over several values and report the best. We have included the effect of the value of hyperparameter β on different subsets of the EgoPER dataset in Figure 1. We observe that a combination of both MOSP and SAI achieves the highest performance.

Ablation on backbone architecture. In Table **??**, we replace the backbone of our proposed method with transformers and replace our whole proposed model with transformers. We observe that our proposed approach with MSTCN backbone achieves a better performance across all metrics. We also compare our approach to GPT zeroshot baseline. In addition to underperforming comaring to our proposed approaches(both zeroshot and trained), using GPT for infering multiple object state prediction directly is not efficient since for each object and state we need to ask the model to make a prediction.

Implementation Details. In order to tackle the oversegmentation issue, we merge predicted segments that have less than 10 background frames in between. This approach gives us smoother predictions.

Baseline Details. For zeroshot baselines CLIP and InternVideo2, we perform a filtering step in which we remove the infeasible objects and their states(for example, we remove brownie when performing inference on a video related to making pizza.) In We get the list of possible objects for each video and possible states for each object from the PLA stage. Therefore, we use action labels groundtruth but not the time segments or ordering.

References

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *International Conference on Machine learning*, 2021.

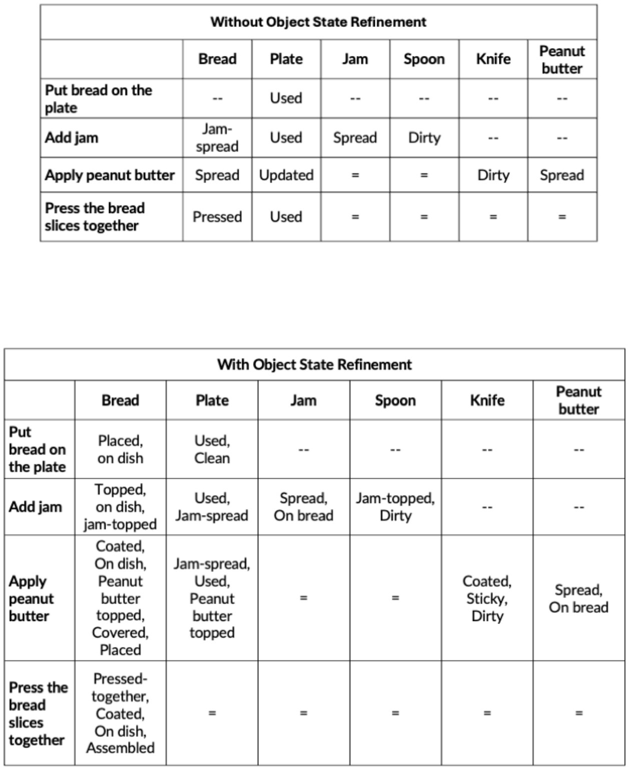
[2] Tomáš Souček, Jean-Baptiste Alayrac, Antoine Miech, Ivan Laptev, and Josef Sivic. Look for the change: Learning ob-

	MOSP					TAS				
	Frame-wise		Segment-wise							
	F1-Max	mAP	F1@0.1	F1@0.25	F1@0.5	Edit	Acc	F1@0.1	F1@0.25	F1@0.5
CLIP [1]	5.97	0.01	0.09	0.03	0.00	2.65	21.68	0.18	0.37	0.91
InternVideo2 [4]	5.66	0.01	0.83	0.66	0.29	8.72	42.87	3.60	6.15	8.02
Pseudo-Labels	11.38	-	**24.24**	**15.15**	**9.09**	-	-	-	-	-
Ours (w/o SAI)	**12.85**	2.57	1.24	0.0	0.0	16.09	29.75	17.41	14.24	5.80
Ours (w/ SAI)	8.04	**2.78**	10.52	6.89	6.45	**21.88**	**40.33**	**22.11**	**17.97**	**9.67**

Table 1. Multiple Object State Prediction and Temporal Action Segmentation results on MOSCATO-EGTEA for the **MOSP** and **TAS** tasks.

	Frame-wise		Segment-wise		
	F1-Max	mAP	F1@0.1	F1@0.25	F1@0.5
GPT	12.10	-	25.77	22.89	16.78
Pseudo-Labels	38.39	-	39.77	35.55	28.73
VidOSC model + Our pseudo-labeling(PLA)	51.23	32.74	52.74	42.49	42.77
Ours (w/o SAI)(MSTCN backbone)	49.99	**35.58**	54.95	46.31	41.07
Ours (w/ SAI)(MSTCN backbone)	**52.83**	35.06	**58.45**	**48.84**	**43.42**
Ours (w/ SAI)(Transformer backbone)	51.54	33.80	56.98	42.85	39.06

Table 2. Ablation studies on backbone architecture on the EgoPER dataset.



Without Object State Refinement						
	Bread	**Plate**	**Jam**	**Spoon**	**Knife**	**Peanut butter**
Put bread on the plate	--	Used	--	--	--	--
Add jam	Jam-spread	Used	Spread	Dirty	--	--
Apply peanut butter	Spread	Updated	=	=	Dirty	Spread
Press the bread slices together	Pressed	Used	=	=	=	=

With Object State Refinement						
	Bread	**Plate**	**Jam**	**Spoon**	**Knife**	**Peanut butter**
Put bread on the plate	Placed, on dish	Used, Clean	--	--	--	--
Add jam	Topped, on dish, jam-topped	Used, Jam-spread	Spread, On bread	Jam-topped, Dirty	--	--
Apply peanut butter	Coated, On dish, Peanut butter topped, Covered, Placed	Jam-spread, Used, Peanut butter topped	=	=	Coated, Sticky, Dirty	Spread, On bread
Press the bread slices together	Pressed-together, Coated, On dish, Assembled	=	=	=	=	=

Figure 2. Comparison of the object state responses from LLM before and after the Stage (3) *Object State Refinement*.

ject states and state-modifying actions from untrimmed web videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Masatoshi Tateno, Takuma Yagi, Ryosuke Furuta, and Yoichi Sato. Learning multiple object states from actions via large language models. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[4] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding. In *Computer Vision – ECCV 2024*, pages 396–416, Cham, 2025. Springer Nature Switzerland.

[5] Zihui Xue, Kumar Ashutosh, and Kristen Grauman. Learning object state changes in videos: An open-world perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18493–18503, 2024.

Datasets		Egocentric	# Obj	# St	# Vid	# StC/Obj	# Obj/Vid	# Hours	Avg. Video Duration(s)	GT/Pseudo
ChangeIt (Training) [2]		×	42	27	34428	1	1	2642	276.0	Pseudo
ChangeIt (Evaluation) [2]		×	42	27	667	1	1	48	276.0	GT
HowToChange (Training) [5]		×	122	20	36075	1	1	395	41.51	Pseudo
HowToChange (Evaluation) [5]		×	134	20	5424	1	1	62.5	41.51	GT
MOST (Training)[3]		×	6	60	10749	> 1	1	1123	336.9	Pseudo
MOST (Evaluation) [3]		×	6	60	61	> 1	1	2.6	156.9	GT
MOSCATO (Training)	CMU	✓	39	77	129	> 1	> 1	13	353.3	Pseudo
	EGTEA	✓	74	79	61	> 1	> 1	23	1083.5	Pseudo
	EgoPER	✓	77	210	163	> 1	> 1	18	150.3	Pseudo
MOSCATO (Evaluation)	CMU	✓	39	77	36	> 1	> 1	3	358.3	GT
	EGTEA	✓	74	79	25	> 1	> 1	6	1781.5	GT
	EgoPER	✓	77	210	50	> 1	> 1	6	148.1	GT

Table 3. Complete verison of comparison of MOSCATO to previous object state change video datasets. # Obj and # St denote the number of objects and states, respectively, in the object vocabulary. # Vid represents the number of videos. # OSC/Vid is the average number of object state changes per video. # StC/Obj is the number of state changes for an object in a video. # Obj/Vid denotes the number of objects that undergo state change in a video. GT/Pseudo determines if that subset is manually (GT) or automatically (Pseudo) labeled.