LSTA: Long Short-Term Attention for Egocentric Action Recognition: Supplementary Document

Swathikiran Sudhakaran[1,2], Sergio Escalera[3,4], Oswald Lanz[1]
[1]Fondazione Bruno Kessler, Trento, Italy
[2]University of Trento, Trento, Italy
[3]Computer Vision Center, Barcelona, Spain
[4]University of Barcelona, Barcelona, Spain
{sudhakaran,lanz}@fbk.eu, sergio@maia.ub.es

This supplementary material provides additional details on the analysis carried out in Sec. 6 of the main manuscript, as well as more visualizations of the attention maps generated by the network.

1. Ablation Analysis

Figs. 1 - 4 show details of the classes which are improved by proposed LSTA variants over the baseline (ConvLSTM) and the difference of the confusion matrices. We show the top 25 improved classes in the comparison graphs and those with less number list all the improved classes. The difference of confusion matrices show the overall details of the classes which are improved. Ideally, the positive values should be in the diagonal and the negative values off-diagonal. Tab. 1 lists a breakdown of the recognition performance. For this, we compute the action recognition and object recognition performance of a network trained for activity recognition. There are some activity classes with multiple objects and these objects are combined to form a meta-object class for this analysis.

Fig. 1 compares the baseline (ConvLSTM) with a network having baseline+output pooling, as explained in Sec. 4.2. It can be seen that adding output pooling to the ConvLSTM improves the network's capability in recognizing different actions with the same objects (`take_water/pour_water,cup` and `close_water/take_water`). This confirms our hypothesis that the output gating of LSTM affects memory tracking, replacing the output gating of LSTM with the proposed output pooling technique localizes the active memory component. This improves the tracking of relevant spatio-temporal patterns in the memory and consequently boosts recognition performance. A gain of 13.79% is achieved for action recognition as shown in Tab. 1.

In Fig. 2, we can see that the network with the attention pooling described in Sec. 4.1 improves the categories with different actions and same objects as well as activity classes with multiple objects (`stir_spoon,cup/pour_sugar,spoon,cup`; `put_cheese,bread/take_bread`; `pour_coffee,spoon,cup/scoop_coffee,spoon`, etc.). Attention helps the network to encode the features from the spatially relevant areas. This allows the network to keep a track of the active object regions and improves the performance. From Tab. 1, a gain of 20.69% is obtained for object recognition which gives further validation regarding the importance of attention.

Adding both attention pooling and output pooling further improves the network's capability in distinguishing between different actions with same objects and same actions with different objects. This is visible in Fig. 3 and also from the 13.72% and 18.1% performance gain obtained for action and object recognition, respectively.

Incorporating bias control, introduced in Sec. 4.2, to the output pooling results in the proposed method, LSTA, which further improves the capacity of the network in recognizing activities (Fig. 4). This further verifies the hypothesis in Sec. 4.2 that bias control increases the active memory localization of the network. This is also evident from Tab. 1 where an increase of 22.41% is obtained for action recognition.

It is worth noting that output pooling boosts action recognition performance more (+13.79% action vs +12,07% object) while with attention pooling the object recognition performance receives a higher gain (+12,93% vs +16,38%). Coupling attention and output pooling through bias control finally boosts performance by a significant margin on both (+22.41% vs +21,55%). This provides further evidence that the two contributions are complementary and reflects the intuitions behind the design choices of LSTA, making the improvements explainable and the benefits of each of the contributions transparently confirmed by

Method	Accuracy (%)		
	Activity	Action	Object
Baseline	51.72	65.52	57.76
Baseline+output pooling	62.07	79.31 (**+13.79**)	69.83 (+12,07)
Baseline+attention pooling	66.38	78.45 (+12,93)	74.14 (**+16,38**)
Baseline+pooling	68.1	79.31 (+13.79)	75.86 (+18,10)
LSTA	74.14	87.93 (**+22.41**)	79.31 (**+21,55**)

Table 1: Detailed ablation analysis on GTEA 61 fixed split. We compute the action and object recognition score by decomposing the action and objects from the predicted activity label.

this analysis.

2. Comparative Analysis

Figs. 5 - 7 compares our method with state-of-the-art alternatives discussed in Sec. 2.3, ego-rnn [2] and eleGatt [3]. Compared to ego-rnn, LSTA is capable of identifying activities involving multiple objects (`pour_mustard,hotdog,bread/pour_mustard, cheese,bread`; `pour_honey,cup/pour_honey, bread`; `put_hotdog,bread/spread_peanut, spoon,bread`, etc.). This may be attributed to the attention mechanism with memory for tracking previously attended regions, helping the network attending to the same objects in subsequent frames. From Fig. 6 it can be seen that eleGAtt-LSTM fails to identify the objects correctly (`take_mustard/take_honey`; `take_bread/take_spoon`; `take_spoon/take_honey`, etc.). This shows the attention map generated by LSTA selects more relevant regions compared to eleGAtt-LSTM .

3. Confusion Matrix

Figs. 8 - 10 show the confusion matrix of the LSTA (two stream cross-modal fusion) for all the datasets explained in Sec. 6.1 of the manuscript. We average the confusion matrices of each of the available train/test splits to generate a single confusion matrix representing the dataset under consideration.

4. EPIC-KITCHENS

We compare the recognition accuracies obtained for EPIC-KITCHENS dataset with the currently available baselines [1] in Tab. 2. As explained in Sec. 6.6 in the paper, we train the network for predicting `verb` and `noun` and `activity` classes. Our two stream cross-modal fusion model obtains an activity recognition performance of 30.33% and 16.63% on S1 and S2 settings as opposed to the 20.54% and 10.89% obtained by TSN strongest baseline (two stream). It is also worth noting that our model is strong on predicting `verb` ($+11.32\%$ points on S1 setting over strongest baseline). This indicates LSTA accurately performs encoding of sequences, indeed `verb` in this context is typically describing actions that develop into an activity over time, and this is learned effectively with LSTA just using video-level supervision.

5. Attention Map Visualization

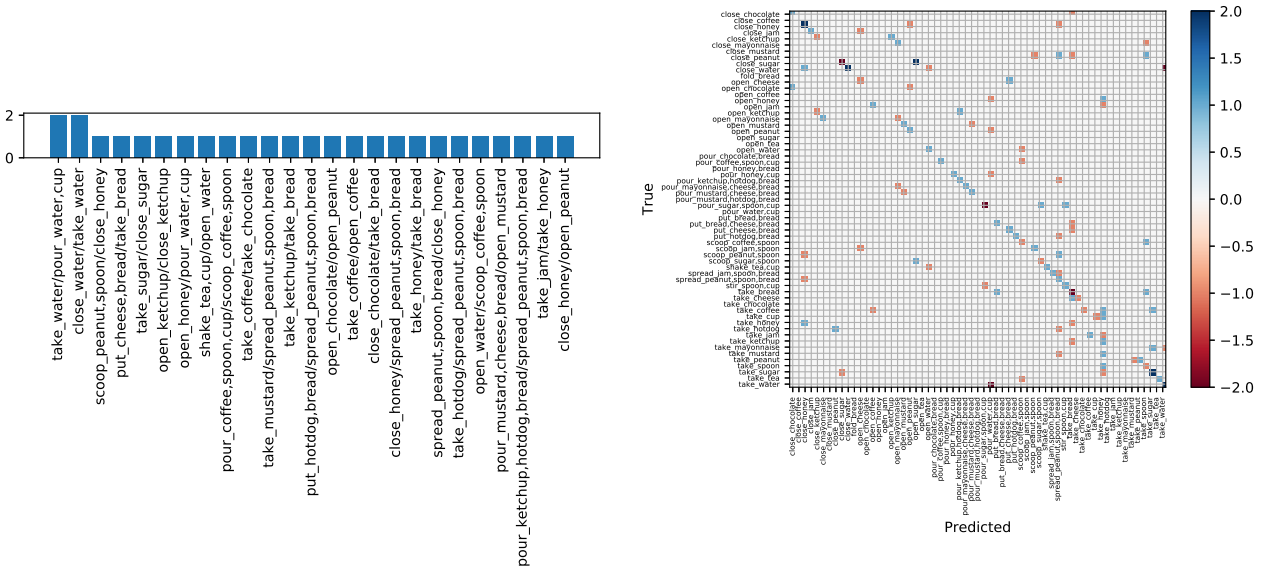
Figs 11 - 15 visualize the generated attention maps for different video sequences. In Figs. 11 - 13, one can see that LSTA is able to successfully identify the relevant regions and track them across the sequences while ego-rnn misses the regions in some frames. This shows the ability of LSTA in identifying and tracking the discriminant regions that are relevant for classifying the activity category. However, in Figs. 14 and 15, the network fails to recognize the relevant regions. In both of these video sequences, the object is not present in the first few frames and the network attends to wrong regions, failing to move its attention towards the object when it appears. Since the proposed method maintains a memory of attention maps, occlusion of the relevant object in the initial frames results in the network attending to the wrong regions in the frame.

References

[1] D. Damen, H. Doughty, G. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proc. ECCV*, September 2018. 2

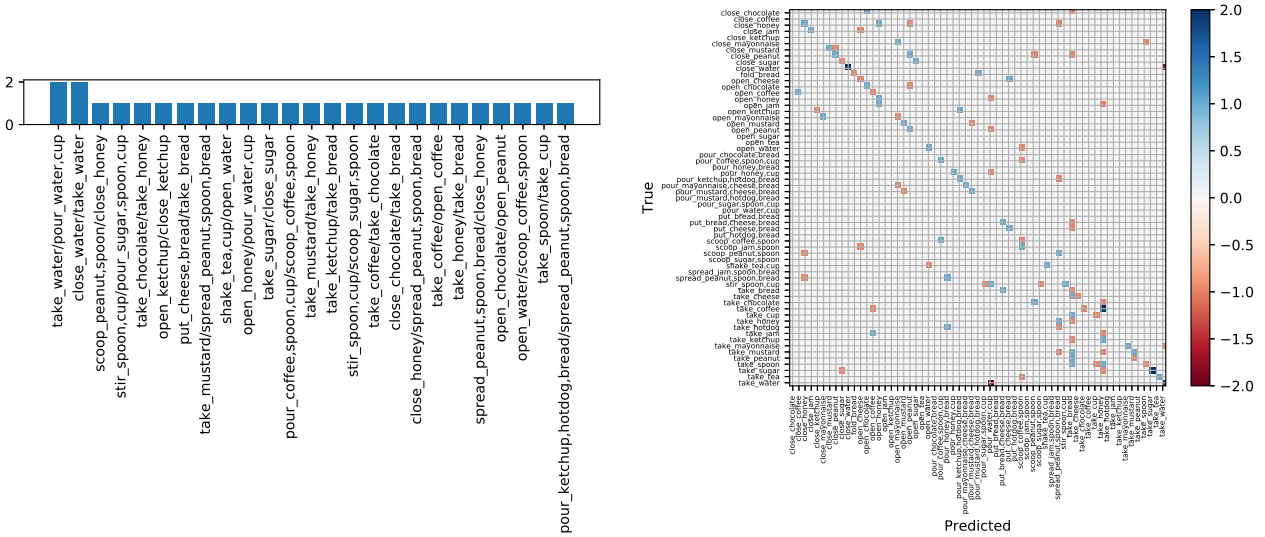
[2] S. Sudhakaran and O. Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *British Machine Vision Conference*, 2018. 2

[3] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng. Adding attentiveness to the neurons in recurrent neural networks. In *Proc. ECCV*, September 2018. 2



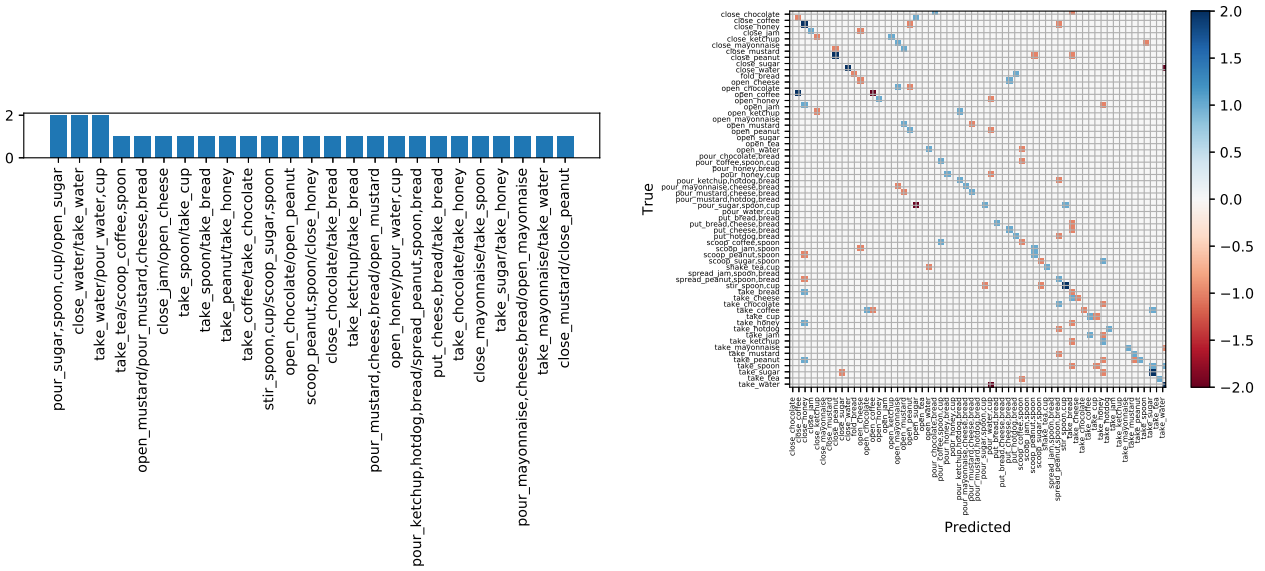
(a) (b)

Figure 1: (a) Most improvement categories by adding output pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + output pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



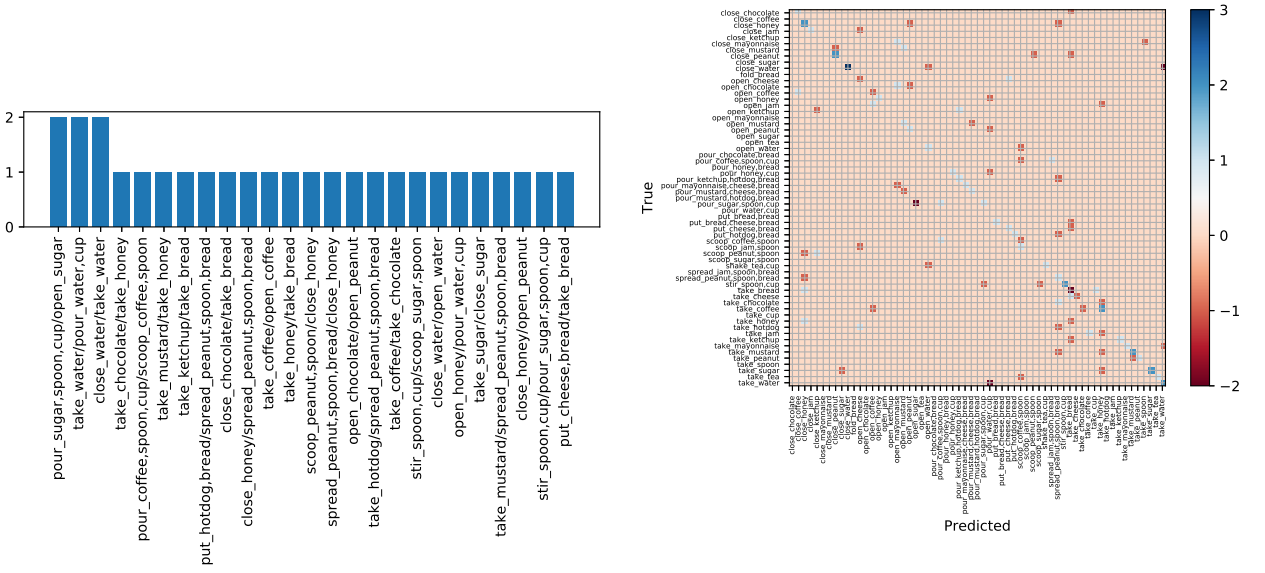
(a) (b)

Figure 2: (a) Most improvement categories by adding attention pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + attention pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



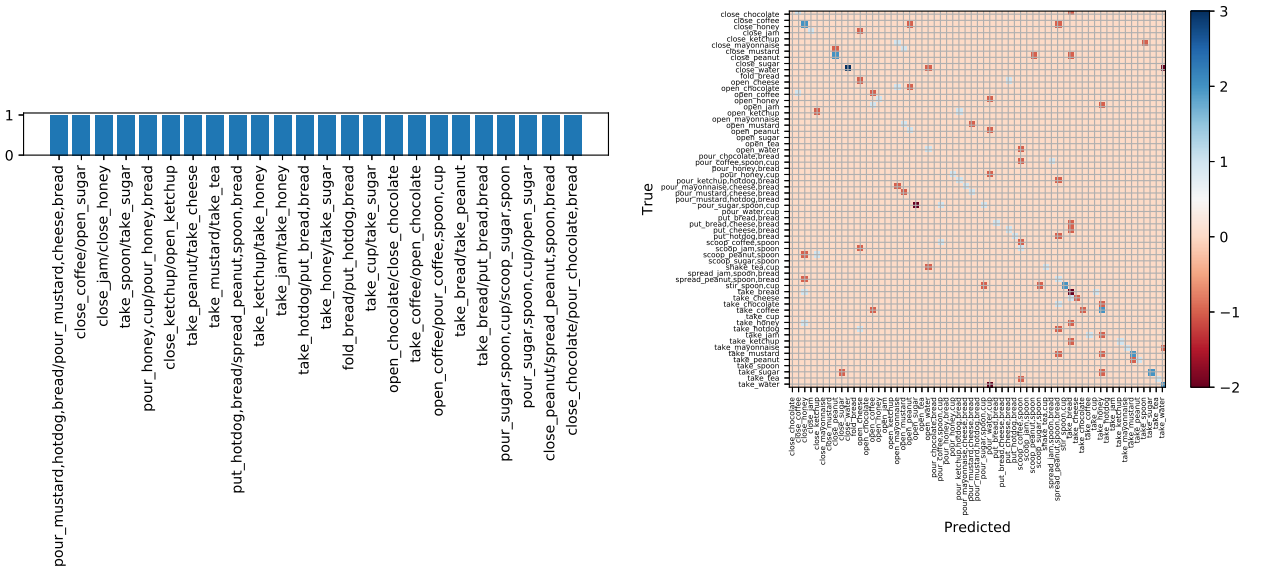
(a) (b)

Figure 3: Most improvement categories by adding both attention and output pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



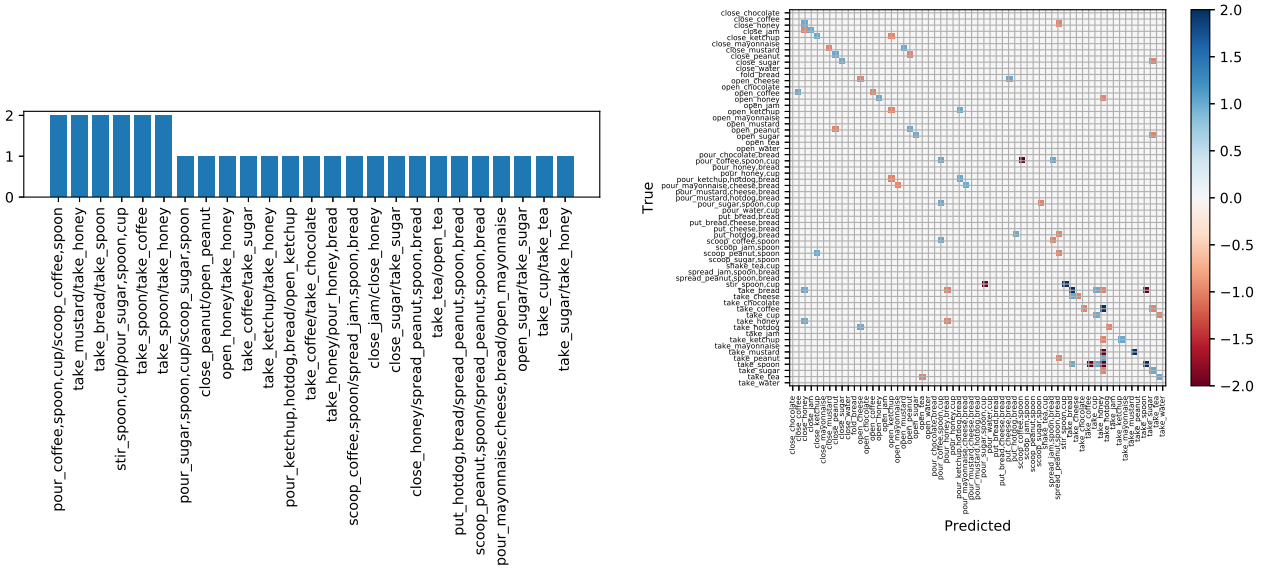
(a) (b)

Figure 4: Most improvement categories by adding attention and output pooling with bias control (full LSTA model) to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



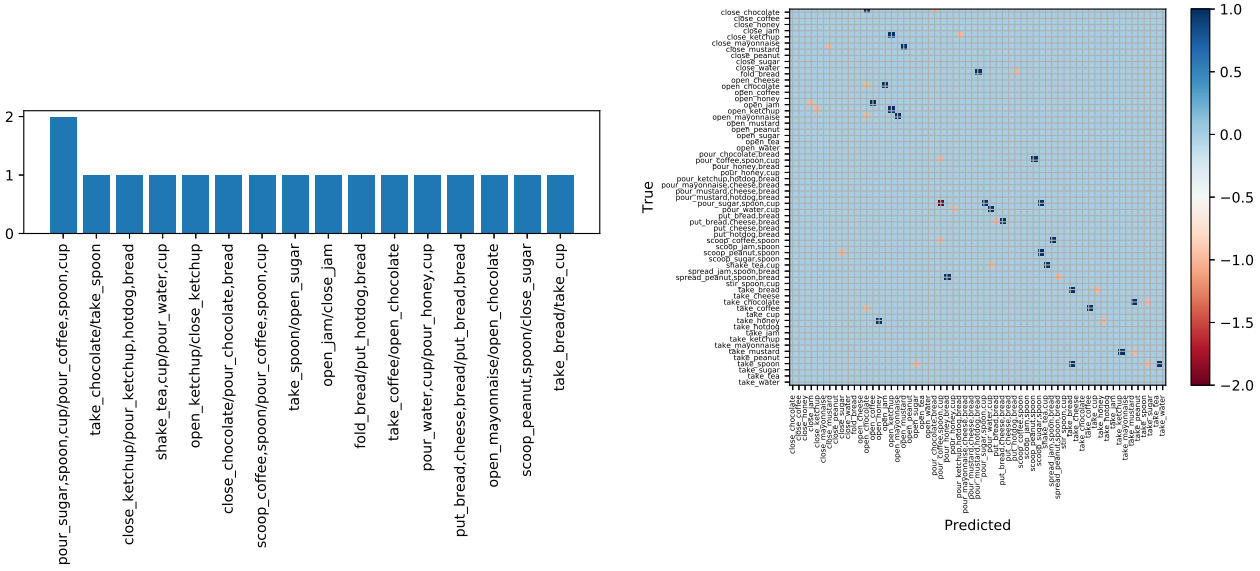
(a) (b)

Figure 5: (a) Most improvement categories by LSTA over ego-rnn on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (ego-rnn). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



(a) (b)

Figure 6: (a) Most improvement categories by LSTA over eleGAtt-LSTM on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (eleGAtt-LSTM). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.



(a) (b)

Figure 7: (a) Most improvement categories by two stream cross-modal fusion over two stream on GTEA 61 fixed split. X axis labels are in the format true label (two stream cross-modal fusion)/predicted label (two stream late fusion). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

	Method	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Precision (%)			Recall (%)		
		Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
S1	2SCNN (RGB)	40.44	30.46	13.67	83.04	57.05	33.25	34.74	28.23	6.66	15.90	23.23	5.47
	2SCNN (two stream)	42.16	29.14	13.23	80.58	53.70	30.36	29.39	30.73	5.92	14.83	21.10	4.93
	TSN (RGB)	45.68	36.80	19.86	85.56	64.19	41.89	**61.64**	34.32	11.02	23.81	31.62	9.76
	TSN (two stream)	48.23	36.71	20.54	84.09	62.32	39.79	47.26	35.42	11.57	22.33	30.53	9.78
	LSTA (RGB)	58.25	**38.93**	30.16	**86.57**	**62.96**	**50.16**	44.09	**36.30**	**16.54**	37.32	**36.52**	**19.00**
	LSTA (two stream)	**59.55**	38.35	**30.33**	85.77	61.49	49.97	42.72	36.19	14.46	**38.12**	36.19	17.76
S2	2SCNN (RGB)	34.89	21.82	10.11	74.56	45.34	25.33	19.48	14.67	5.32	11.22	17.24	6.34
	2SCNN (two stream)	36.16	18.03	7.31	71.97	38.41	19.49	18.11	15.31	3.19	10.52	12.55	3.00
	TSN (RGB)	34.89	21.82	10.11	74.56	45.34	25.33	19.48	14.67	5.32	11.22	17.24	6.34
	TSN (two stream)	39.4	22.7	10.89	74.29	**45.72**	25.26	22.54	15.33	6.21	13.06	17.52	6.49
	LSTA (RGB)	45.51	**23.46**	15.88	75.25	43.16	30.01	26.19	17.58	8.44	20.80	**19.67**	11.29
	LSTA (two stream)	**47.32**	22.16	**16.63**	**77.02**	43.15	**30.93**	**31.57**	**17.91**	**8.97**	**26.17**	17.80	**11.92**

Table 2: Comparison of recognition accuracies with state-of-the-art in EPIC-KITCHENS dataset.

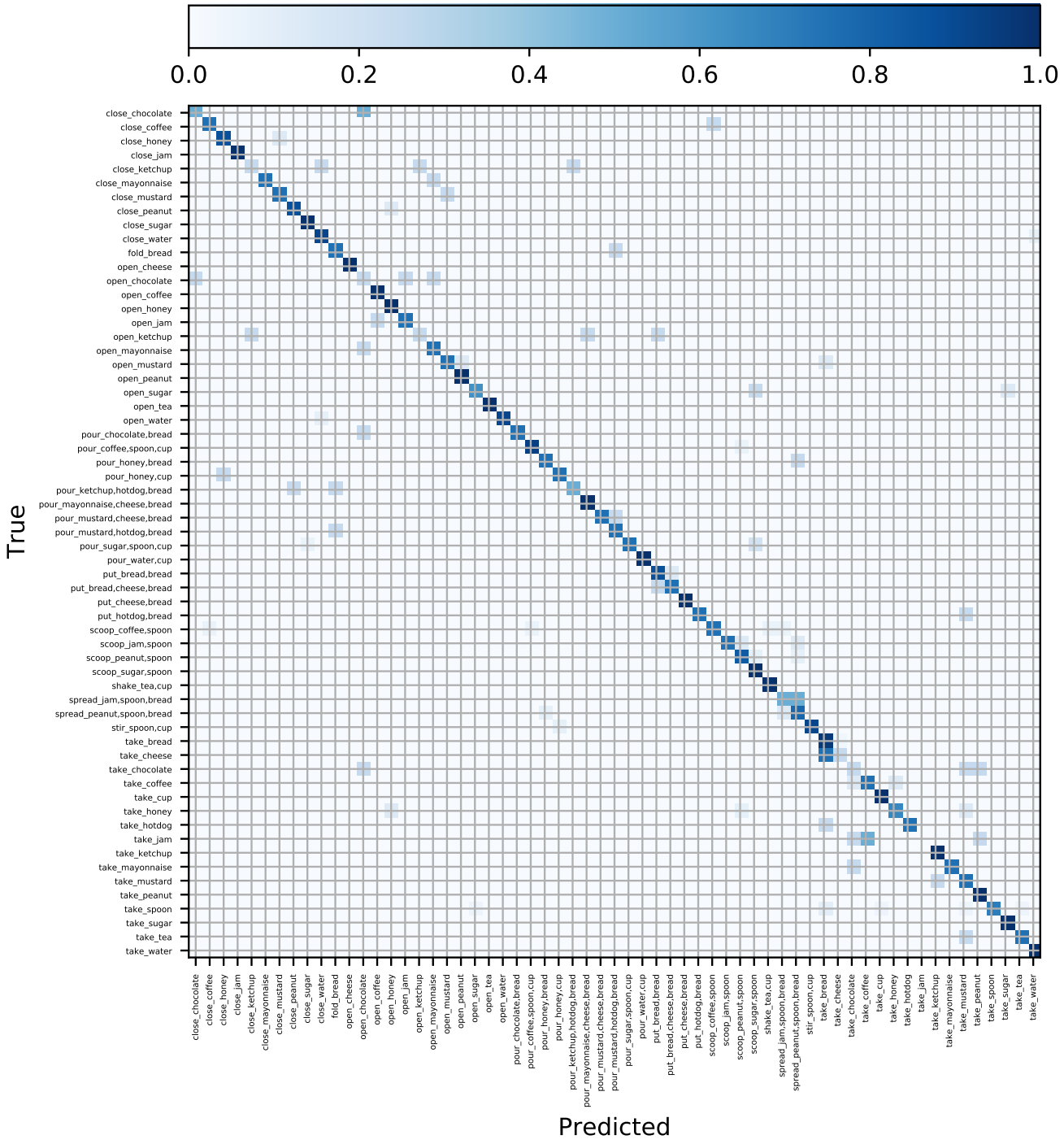


Figure 8: Confusion matrix of GTEA 61 averaged across the four train/test splits.

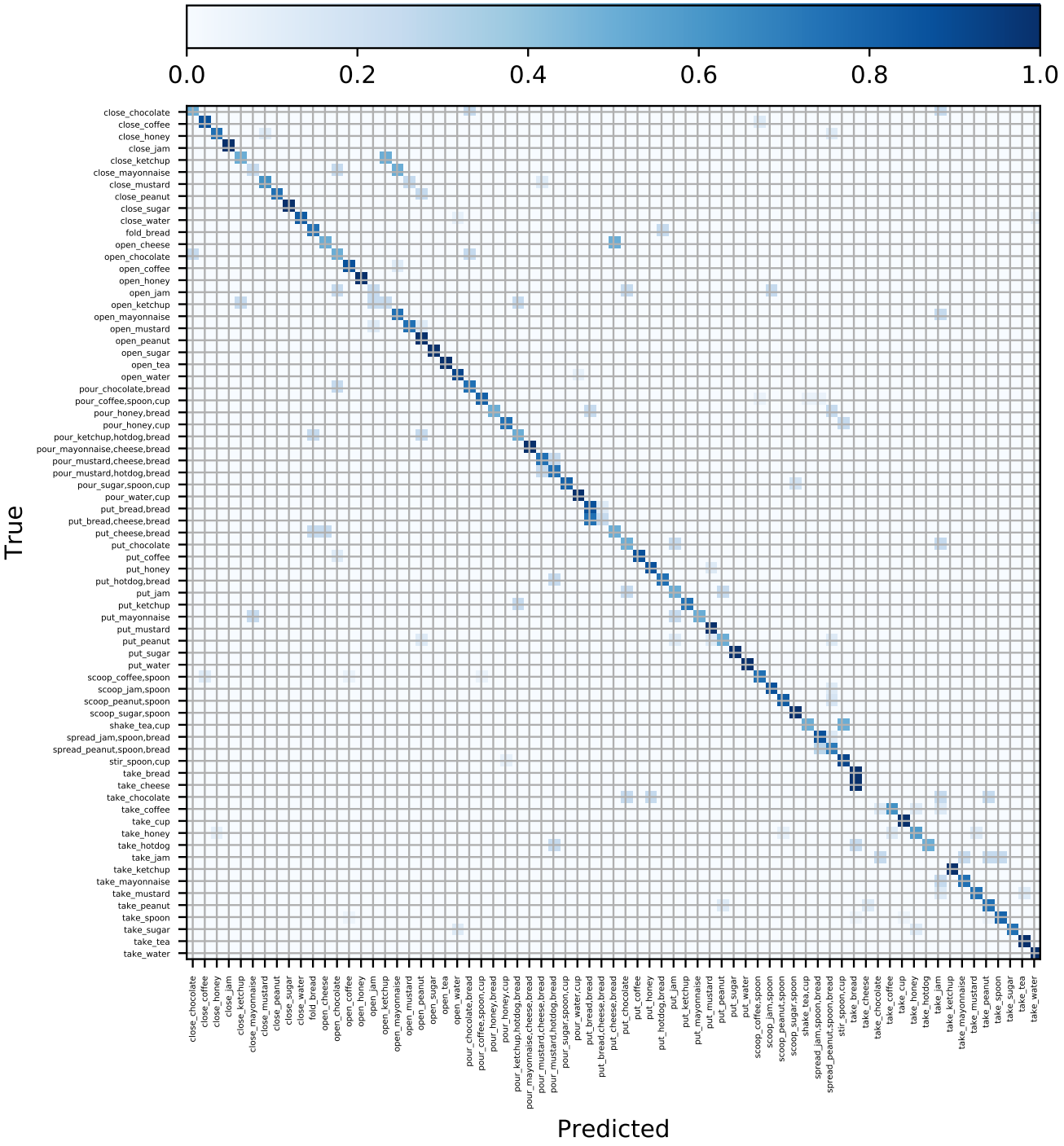


Figure 9: Confusion matrix of GTEA 71 averaged across the four train/test splits.

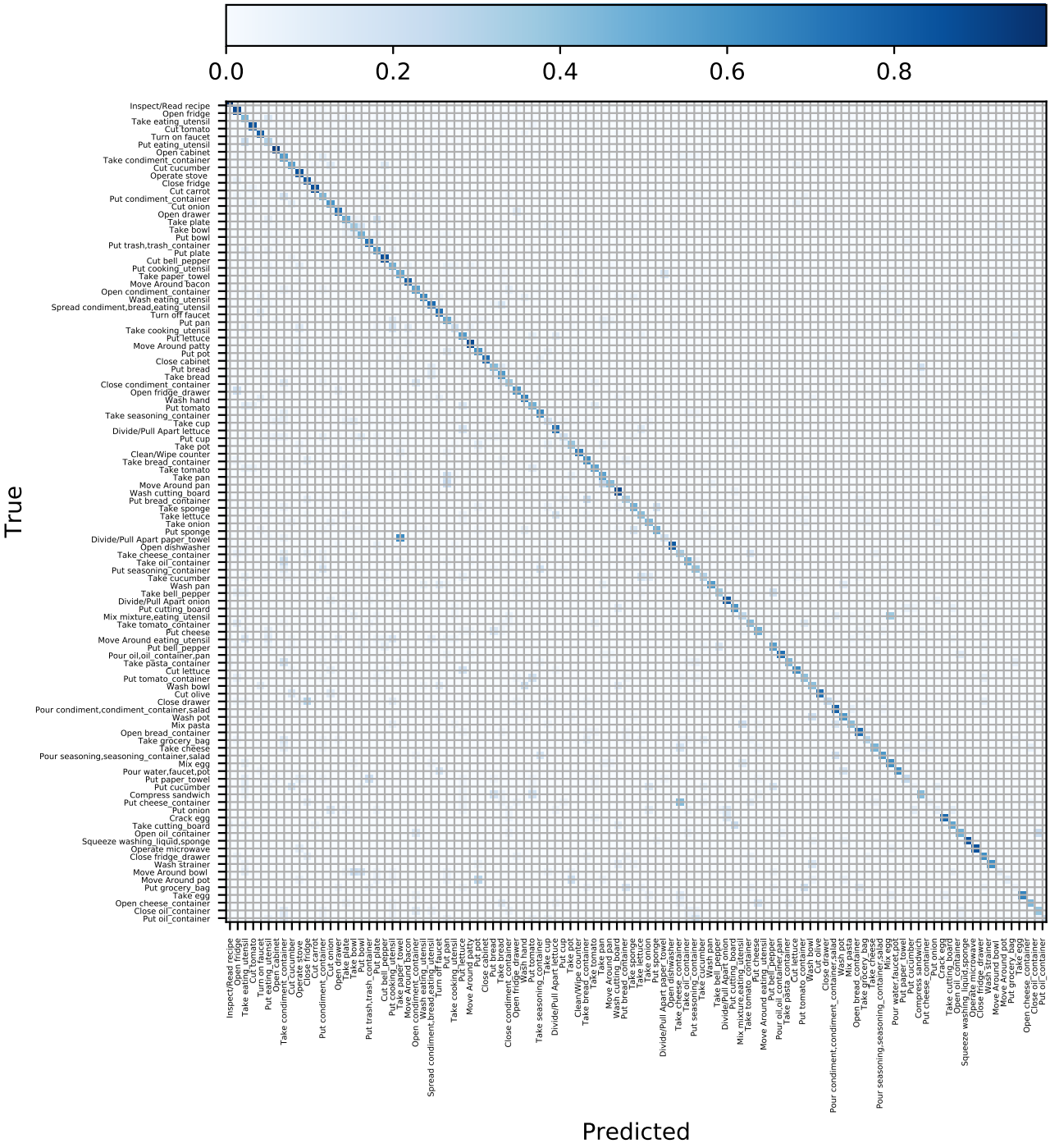


Figure 10: Confusion matrix of EGTEA Gaze+ averaged across the three train/test splits.

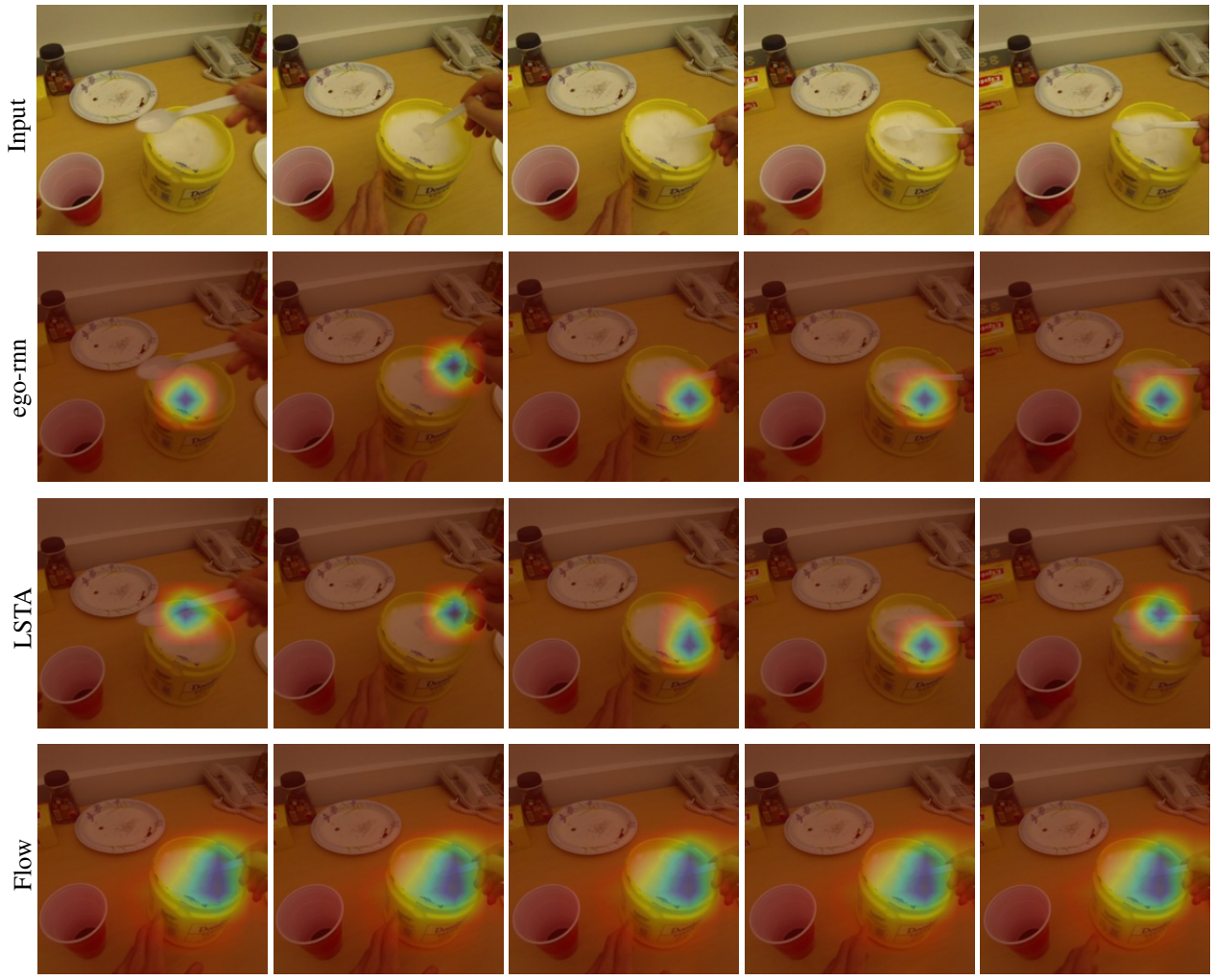


Figure 11: Attention maps generated by ego-rnn (second row) and LSTA (third) for scoop_sugar,spoon video sequence. We show the 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks. Fourth row shows the attention map generated by the motion stream. For flow, we visualize the attention map on the five frames corresponding to the optical flow stack given as input.

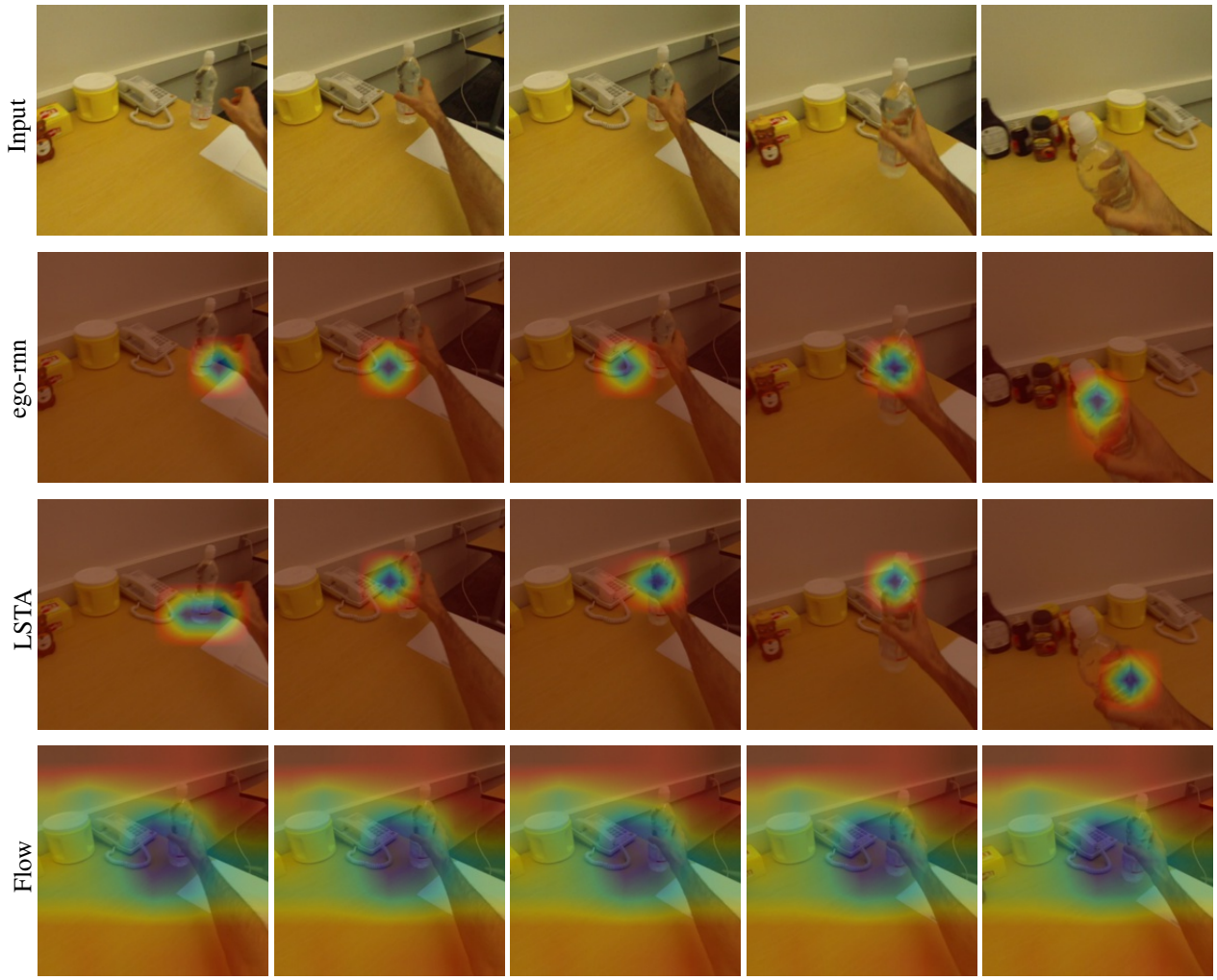


Figure 12: Attention maps generated by ego-rnn (second row) and LSTA (third) for take_water video sequence. We show the 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks. Fourth row shows the attention map generated by the motion stream. For flow, we visualize the attention map on the five frames corresponding to the optical flow stack given as input.

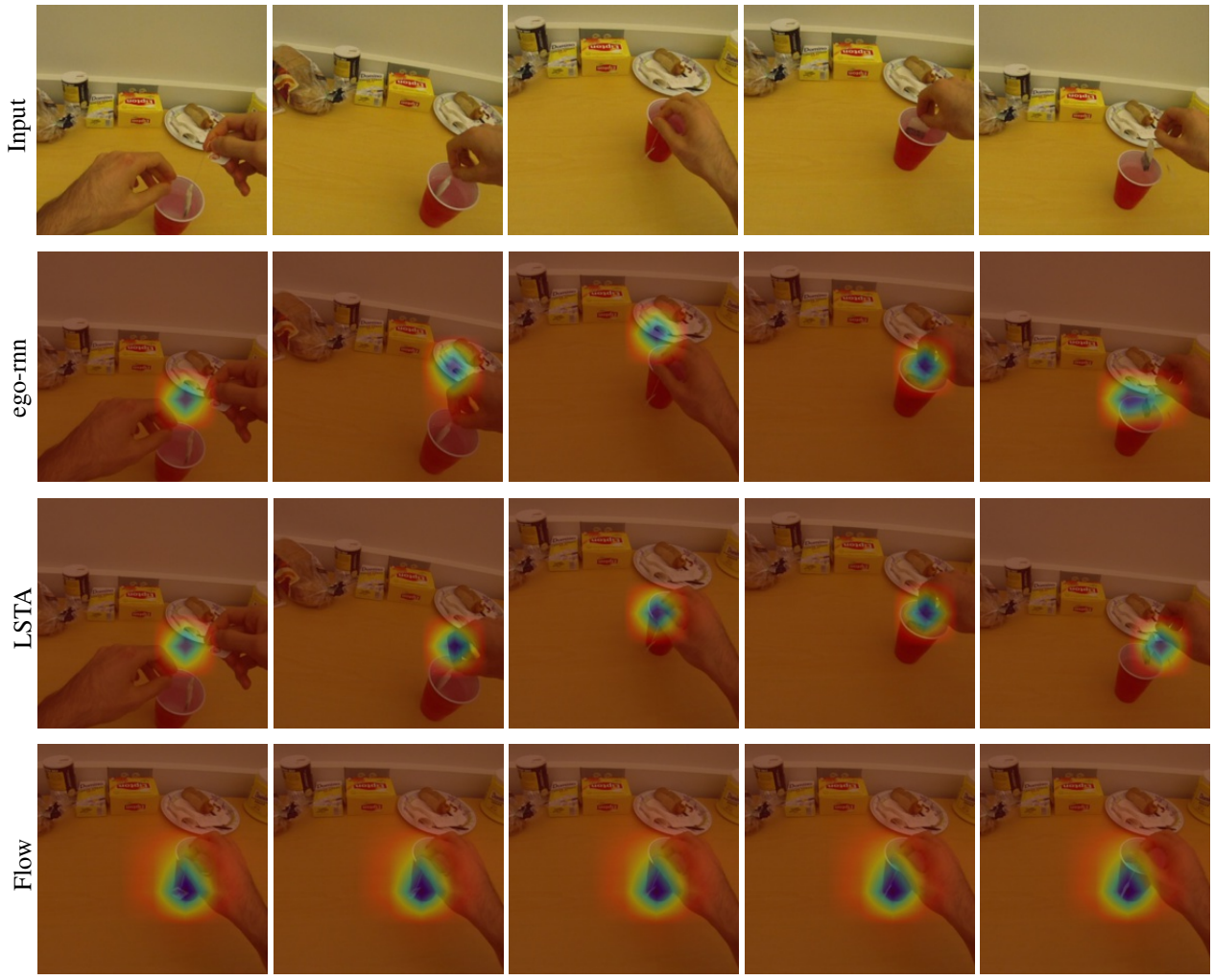


Figure 13: Attention maps generated by ego-rnn (second row) and LSTA (third) for shake_tea,cup video sequence. We show the 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks. Fourth row shows the attention map generated by the motion stream. For flow, we visualize the attention map on the five frames corresponding to the optical flow stack given as input.

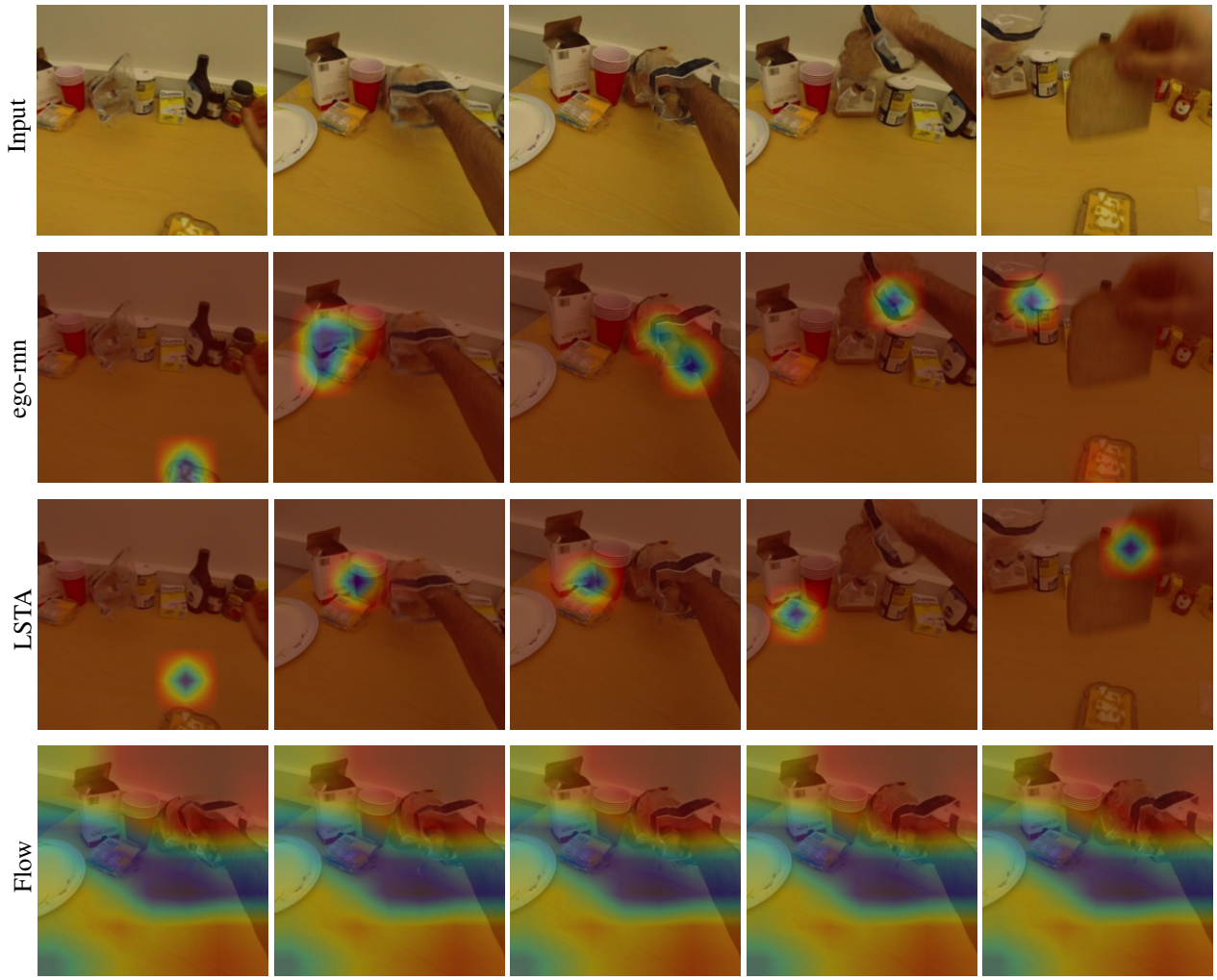


Figure 14: Attention maps generated by ego-rnn (second row) and LSTA (third) for take_bread video sequence. We show the 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks. Fourth row shows the attention map generated by the motion stream. For flow, we visualize the attention map on the five frames corresponding to the optical flow stack given as input.

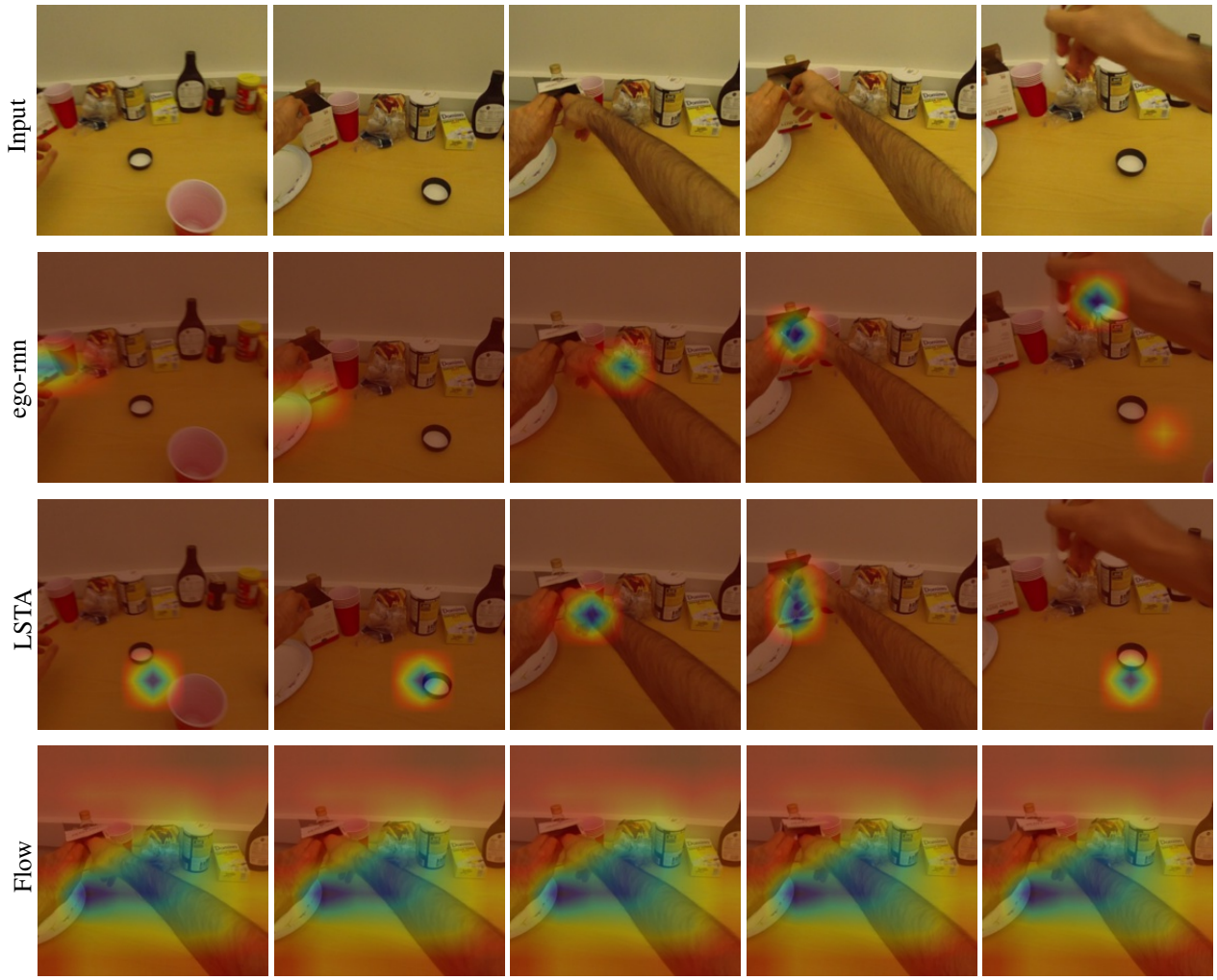


Figure 15: Attention maps generated by ego-rnn (second row) and LSTA (third) for take_spoon video sequence. We show the 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks. Fourth row shows the attention map generated by the motion stream. For flow, we visualize the attention map on the five frames corresponding to the optical flow stack given as input.