Robot Learning and Execution of Collaborative Manipulation Plans from YouTube Videos

Hejia Zhang and Stefanos Nikolaidis

***Abstract*— People often watch videos on the web to learn how to cook new recipes, assemble furniture or repair a computer. We wish to enable robots with the very same capability. This is challenging; there is a large variation in manipulation actions and some videos even involve multiple persons, who collaborate by sharing and exchanging objects and tools. Furthermore, the learned representations need to be general enough to be transferable to robotic systems. Previous systems have enabled generation of semantic and human-interpretable robot commands in the form of visual sentences. However, they require manual selection of short action clips, which are then individually processed.**

We propose a framework for executing demonstrated action sequences from full-length, unconstrained videos on the web. The framework takes as input a video annotated with object labels and bounding boxes, and outputs a collaborative manipulation action plan for one or more robotic arms. We demonstrate the performance of the system in three full-length collaborative cooking videos on the web and propose an open-source platform for executing the learned plans in a simulation environment.

I. INTRODUCTION

We focus on the problem of learning collaborative action plans for a robot. Our goal is to have the robot "watch" unconstrained videos on the web, extract the action sequences shown in the videos and convert them to an executable plan that it can perform either independently, or as part of a human-robot or robot-robot team.

Learning from online videos is hard, particularly in collaborative settings: it requires recognizing the actions executed, together with manipulated tools and objects. In many collaborative tasks these actions include handing objects over or holding an object for the other person to manipulate. There is a very large variation in how the actions are performed and collaborative actions may overlap spatially and temporally[1].

In our previous work [2], we proposed a system for learning activities performed by two humans collaborating at a cooking task. The system implements a collaborative action grammar built upon the action grammar initially proposed by Yang et al. [3]. A qualitative analysis in 12 clips showed that parsing these clips with the grammar results in human-interpretable tree structures representing a variety of single and collaborative actions. The clips were manually segmented and were approximate 100 frames each.

In this paper, we generalize this work with a framework for *generating single and collaborative action trees from full-length YouTube videos lasting several minutes* and *concatenating the trees in an action graph which is executable by one or more robotic arms*.

Hejia Zhang and Stefanos Nikolaidis are with the Department of Computer Science, University of Southern California, Los Angeles, USA {hejiazha, nikolaid}@usc.edu

Fig. 1: The robots execute the action sequence shown in the video.

The framework takes as input a YouTube video showing a collaborative task from start to end. We assume that the objects in the video are annotated with labels and bounding boxes, e.g., by running a YOLOv3 algorithm [4] (Fig. 1). We also assume a skill library that associates a detected action with skill-specific motion primitives. We focus on cooking tasks because of the variety in manipulation actions and their importance in home service robotics.

Fig. 2 shows the components of the proposed framework. We rely on the insight that hands are the main driving force of manipulation actions [5]. We detect the human hands in the video and use the hand trajectories to split the video into clips. We then associate objects and hands spatially and temporally to recognize the actions and generate human-interpretable robot commands. Finally, we propose an open-sourced platform for generating and executing action graph. We provide a quantitative analysis of performance in two YouTube videos of 13401 frames in total and a demonstration in simulation of robots learning and executing correctly the actions of a third video of 2421 frames.

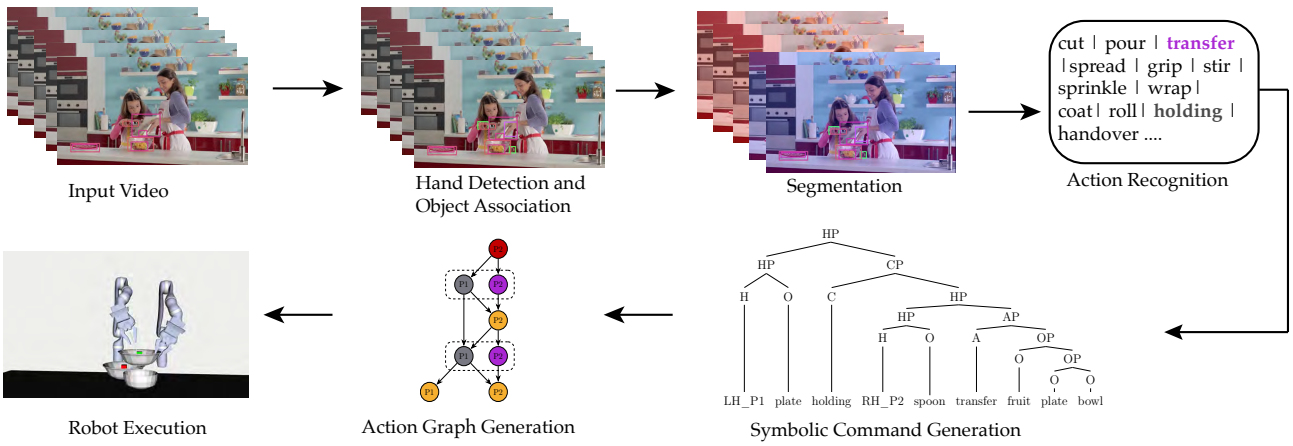


Fig. 2: Proposed framework architecture.

While the extracted action sequences are executed in an open-loop manner and thus do not withstand real-world failures or disturbances, we find that this work brings us a step closer to having robots generate and execute a variety of semantically meaningful plans from watching videos online.

II. RELATED WORK

In this paper we propose a framework for converting full-length unconstrained videos from the Internet to collaborative action plans executed by one or more robots. Most relevant to ours is prior work on temporal video segmentation and action understanding.

A. Temporal Video Segmentation

Work in video segmentation includes learning Gaussian Mixture Models [6], detecting changepoints through filtering and smoothing techniques [7] and specifying cost-functions incorporating spatial and temporal features of the trajectory [8]. When the output of a supervised learning algorithm is available, particle filter-based sampling approaches can integrate predictions to infer a sequence of activity classes [9]. Recent work also combines classifier outputs with a symbolic grammar to parse sequence data [10]. In the work by Lioutikov et al. [11], movement primitives are learned in conjunction with trajectory segments, using an iterative Expectation-Maximization (EM) algorithm. In this paper we apply the Greedy Gaussian segmentation algorithm by Hallac et al. [12]. While the algorithm is applicable for general multivariate time-series data, we adopt it for segmenting videos to action clips using hand trajectories, based on the insight that hands are the main driving force of manipulation actions.

B. Action Understanding

There has been a lot of work on human activity recognition [13]. Recent work on deep learning approaches has enabled the generation of natural language [14], individual robot commands [15], and neural programs [16] using manually annotated datasets. Generation of collaborative actions has been achieved by representing them as social affordances [17] extending previous work on object affordance learning [18], [19] or as interaction primitives [20] from data recorded in a lab setting. Generalization is an important challenge in robot learning and to address this issue, Pastra et al. [21] discuss a minimalist grammar for action understanding, inspired by the suggestion by Chomsky [22]. An implementation of such a grammar for activity understanding was provided by Summers-Stay et al. [23]. The probabilistic manipulation action grammar was first proposed by Yang et al. [3], [5]. The system uses deep neural networks for hand and object detection and association, while it leverages a language corpus for action recognition. We have recently extended the grammar to account for collaborative tasks for a robot interacting with a human teammate [2]. In these works [3], [2], the grammar was used to generate action trees from 12 selected video clips, rather than from full-length unconstrained videos.

III. FRAMEWORK

The input to the framework is a full-length, unconstrained video from the web. We assume that objects in the video are labeled and a bounding box is provided for each object e.g., using a state-of-the-art object detection algorithm [4]. We base this assumption on the tremendous progress of recent object-detection algorithms and the availability of large datasets. This is the only labeled input data provided to the framework.[1]

A. Hand Detection

Our work relies on the insight that hands are the main driving force of manipulation actions [5]. We use OpenPose [24], which detects jointly the human body and hands. We use the detected hands to (1) segment videos by tracking the hand trajectory, and (2) detect which objects are manipulated at a given point in time.

[1] The current implementation of the framework works for videos of one person working independently or two persons collaborating. An extension to three or more persons interacting in pairs is straightforward and left for future work.

B. Video Segmentation

We temporally segment the video to short clips using the trajectories of the detected hands as time-series data, performing a separate segmentation for each hand of the actors in the video. We use a greedy approach [12], which formulates the segmentation as a covariance-regularized maximum likelihood problem of finding the segment boundaries.

We then generate a new sequence of segments for the whole video as the union of individual segments, that we will use for action recognition. Based on the assumption that actions require at least 1 second to be executed, we filter out segments that are shorter.

This method results in over-segmentation with some actions spanning multiple segments, which is common in segmentation algorithms [25]. We also have several segments that do not include any action. Therefore, we merge segments in the action graph generation phase (Section III-F).

C. Object Association

After video segmentation, we extract objects that are relevant to actions in each segment. We do this by associating objects with hands and with other objects based on their relative positions in the frame. We extend previous work [2] by introducing a semantic hierarchy of objects based on commonsense reasoning. Specifically, we assign objects to three classes: *tools* that manipulate other objects, e.g., knife and fork, *containers* that can "contain" other objects, e.g., pot and bowl, and *ingredients*, e.g., banana and lemon, that can not contain other objects. For robustness, we only keep the hand and object associations retained for a minimum number of consecutive frames.

Hand-Object association. We want to detect the objects grasped by the hands and then propagate this association to nearby objects that can inform the action recognition. This allows us to infer which objects are directly manipulated or used as tools to manipulate other objects.

We associate detected hands with objects whose bounding boxes overlap with the box of the hand. In the case of multiple overlaps, we associate the hand with the container or tool that has the largest overlap. If there is no such object, we associate it with the nearest ingredient. We look first for tools and containers, since they are larger and thus make associations more robust.

Object-Object association. For each object that has been associated with a hand, we look to associate that object with other objects that are possibly manipulated. For instance, if a hand grasps a spoon, we wish to see if the spoon is used to stir a pot nearby. As in hand-object association, we look first for the nearest container that has an overlapping bounding box and then for the nearest ingredient if there is no container nearby.

We finally associate container objects with ingredients, if there is an overlap in the bounding boxes of the two. We use containers to detect transfer of objects from one container to the other, e.g, transfer a tomato from a bowl to a chopping board. We use the Jaccard index [26] between the bounding boxes of the two objects to pair a container with one or more ingredient objects.

D. Action Recognition

After segmenting the video into clips and pairing objects with hands, we recognize actions performed by humans in the videos. We have two types of actions, actions performed by a single person, which we name *individual actions*, and *collaborative actions* performed by a pair of humans in the video. As a special case of individual actions, we introduce *transfer* actions, which occur when an object moves from one container to another. This allows to detect transfer of an ingredient between containers.

Individual Actions. Following the approach by Yang et al. [3], we recognize "commonsense actions" using a trained language model from a general purpose language corpus [27] and a recipe corpus [28]. Given a set of candidate actions and a set of candidate objects, we extract $P(Object|Action)$ for each possible bigram consisting of one object word and one action word in corpus. We then compute the probabilities of each action given the involved objects such as tool used, ingredient manipulated as follows:

$$P(A|O_1, O_2, O_3) \sim P(O_1|A)P(O_2|A)P(O_3|A)P(A)$$

where A is the performed action and O_1, O_2, O_3 are the tool, ingredient and container involved in the action respectively. We then select the most likely action. We use the general corpus for the container and tool - action bigrams and the action prior, and the recipe corpus for the ingredient bigrams.

Transfer Actions. We treat transfer actions separately from the other individual actions, since they occur when an object is moved from one container to the next and thus require tracking an object's association temporally. These actions are critical in keeping track of the location of the food in the cooking task.

Collaborative Actions. Following previous work [2], we detect a collaboration: (1) when two persons grasp the same object, or (2) the object grasped by one person is used as a tool to manipulate an object grasped by another person. In case (1), we check over time which hands grasp the object and detect a *handover* if the person grasping the object changes. Otherwise, we detect a *holding* action, for instance when one person assists the other person stirring a pot by holding the pot as well.

E. Action Grammar Parsing

We need to represent the structure of the recognized actions for a robot to execute them. We use a manipulation action grammar [3] which assumes that hands (H) are the driving force of both single manipulation actions (A) and collaborative actions (C). A hand phrase HP contains an action phrase AP, or a collaborative action phrase CP. We extend the collaborative action grammar of previous work [2] by introducing an object phrase OP. We use the object phrase to indicate container - ingredient relationships

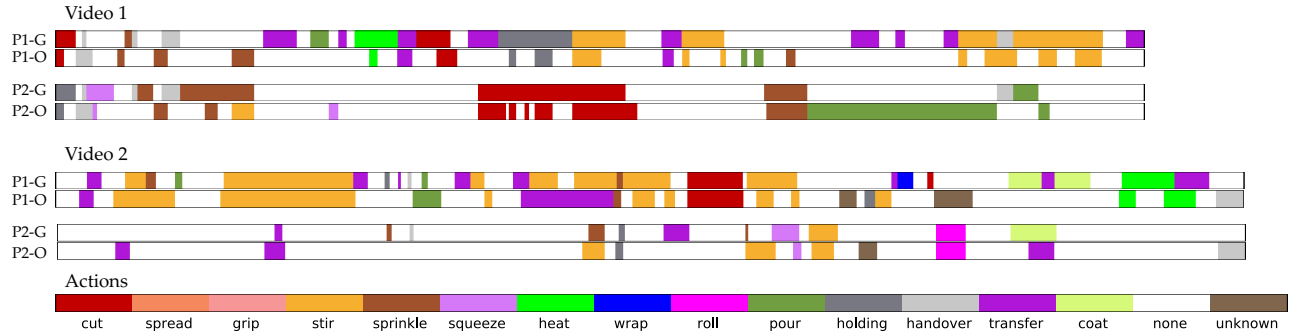


Fig. 3: Segmentation of the two test videos. For each video we show the groundtruth for the woman (P1-G), our result for the woman (P1-O), the grountruth for the girl (P2-G) and our result for the girl (P2-O). The bottom color bar shows different colors for different actions. The generated segments are merged together afterwards in a post-processing phase.

$$HP \rightarrow H\ O \mid HP\ AP \mid HP\ CP \quad (1)$$
$$AP \rightarrow A\ O \mid A\ OP \mid A\ HP \quad (2)$$
$$CP \rightarrow C\ HP \quad (3)$$
$$OP \rightarrow O\ O \mid O\ OP \quad (4)$$
$$H \rightarrow Hand \quad (5)$$
$$C \rightarrow Collaboration \quad (6)$$
$$O \rightarrow Object \quad (7)$$
$$A \rightarrow Action \quad (8)$$

Fig. 4: A Collaborative Manipulation Action Context-Free Grammar

between objects, e.g. a tomato in the bowl, as well as transfer actions from one container to another. The grammar is given in Fig. 4. The rules (5)-(8) are terminal, with *Hand* taking the values: "LH_P1", "RH_P1", "LH_P2" and "RH_P2," "LH_P1" being the left hand of the first person and so on. We use a context-free grammar parser [29] to parse the constructed visual sentences [2] and output a parse tree of the specific manipulation action. The robot can then execute the action by reversely parsing the tree. Fig. 7 shows the constructed trees from different action clips.

F. Action Graph Generation and Execution

Because of over-segmentation (Section III-B), we end up with multiple consecutive segments that are parts of the same action. Therefore, we first merge consecutive segments from the video with identical actions, hand-object and object-object associations. We do not require identical ingredient objects, since they may not be visible in some of the segments.

We then generate an action graph that combines the generated action trees to action sequences, each corresponding to each person in the video. We then decompose each action into motion primitives. We define four primitives [30]: grasp, engage, actuate and place. For instance, a transfer action of a food from a plate to a bowl with a spoon includes grasping the spoon (grasp), moving it close to the food (engage), performing the scooping motion (actuate), moving the spoon close to the bowl (engage), turning it to remove the food (actuate), and placing it back in its initial position (place). We use Task Space Regions (TSRs) [31] to specify feasible regions of target poses of the robot's end effector in the grasp, engage and place primitives, and we use bidirectional rapidly-exploring random trees (BiRRT) [32] to plan collision-free paths.

The action graph enables transitioning and ordering in both the task action and motion primitive levels:

Transitioning. People often grasp an object and use it as a tool for consecutive actions. We enable smooth transitioning of two consecutive actions with the same tool by removing the place and grasp motion primitives of these actions.

Ordering. The action graph ensures that the actions of each person are executed in the demonstrated order. Additionally, a collaborative action is executed only when both agents have reached the corresponding node in the graph. In the lowest level, the action graph ensures ordering of motion primitives in collaborative actions (e.g., wait until the other agent has engaged before actuating in a handover).

We implement the action graph as an open-source platform, named WeCook, that enables collaborative task execution in the cooking domain.[2] It is based on AIKIDO [33], a C++ library for robotic motion planning and decision making.

IV. EXPERIMENTS

We demonstrate the applicability of our framework in two unconstrained YouTube videos of two persons doing a collaborative cooking task together.[3,4] We set up a start and end time for each video, annotated the objects and set bounding boxes. We only annotated objects that were clearly visible, skipping objects that were not rigid (e.g., water) or heavily occluded. The videos included a total of 13401 frames and 67 actions of 12 different action types.

Fig. 3 shows the result of the segmentation and action recognition in the two videos before merging the segments

[2] https://github.com/icaros-usc/wecook

[3] https://www.youtube.com/watch?v=jAhQfH1PspU&t=119s

[4] https://www.youtube.com/watch?v=1p2wBBmhPmk&t=138s

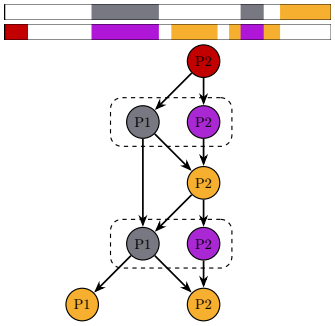


Fig. 5: The video segmentation result and generated task graph. Different actions are shown in different colors using the colorbar of Fig. 3. P_1 indicates an action executed by the woman in the video and P_2 by the girl. The dotted line indicates a collaboration between the two actors.

(Section III-F). We can see clearly that the woman in the video performed most of the actions, especially in video 2.

After merging the segments, we evaluate the performance of the framework with respect to the percentage of correctly learned action-trees and actions. We define a correct action tree when the structure and all nodes of the tree are identical to the ground-truth, and the segment corresponding to that tree has a non-zero temporal overlap with the ground-truth segment. We specify the *precision* as the number of action trees the framework returns correctly out of the total number of detected instances, and the *recall* as the number of action trees the framework returns correctly out of the total number of ground-truth trees.

The precision and recall for action-trees were 0.60 and 0.44, while for action recognition only (ignoring correctness of all nodes in the tree) they were 0.63 and 0.46. This indicates that most action trees generated by the system are correct, while some ground-truth actions are hard to detect and they are missed by the system.

Fig. 7 shows the action-trees and snapshots of the action tree executions by two robotic arms in the WeCook platform. The proposed method reproduces successfully a variety of individual and collaborative actions.

V. DEMONSTRATION

To demonstrate the applicability of our framework, we selected an "easy" video of 2421 frames,[5] where our framework achieved perfect segmentation and action recognition (Fig. 5). In the accompanying video,[6] we show the execution of the complete action graph by two simulated Kinova Gen2 lightweight [34] robotic arms in WeCook.

VI. DISCUSSION

Limitations. Our work is limited in many ways. Some failures are caused by our assumptions in object detection and associations. Fig. 6 illustrates different failure cases. In Fig. 6(a), we did not include the chopped onion in the object annotations, since it was not clearly visible. Therefore,

[5]https://www.youtube.com/watch?v=d3SZH7NFDjc&list=PL4C3C1C9AB9931360&index=75

[6]https://www.youtube.com/watch?v=iNN5q-IHiJg&feature=youtu.be

No tree generated

(a) The girl is sprinkling onion to the pot.

(b) The woman is pouring oil to the pot.

(c) The woman is not performing any action.

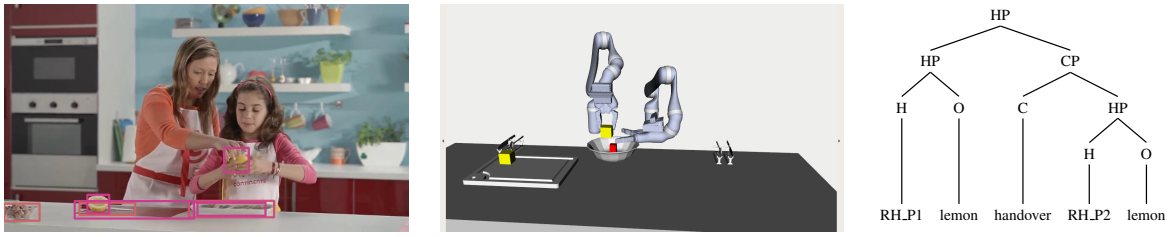
Fig. 6: Example frames and generated trees of 3 failure cases. The captions depict the groundtruth descriptions of each case.

the system failed to generate an action tree for the sprinkle action. In Fig. 6(b), the closest container object to the oil is pan, while the woman is actually pouring oil to the pot. In Fig. 6(c), the system incorrectly infers that the woman is grasping the oil and associates it with the pot object, although the woman's hand does not make actual contact with the oil.

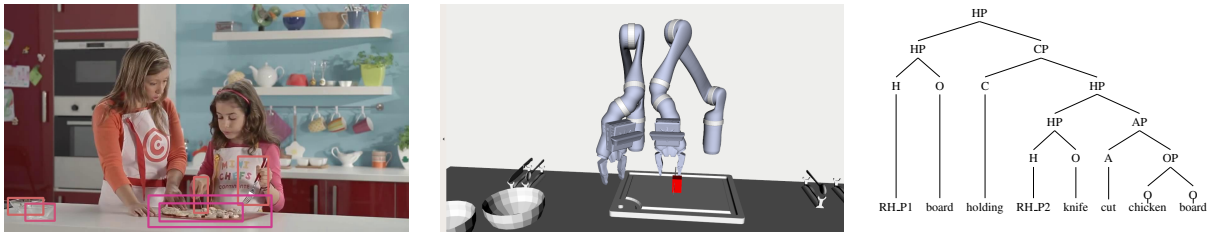
Additionally, although the selected YouTube videos were unconstrained, they were meant to be instructional and thus relatively clear. While our hand and object detection-based action prediction can be robust against implicit or ambiguous actions, language corpus-based commonsense reasoning will fail in infrequent cases, such as cutting food on a bowl with a spoon instead of a knife. Learning embeddings from cooking recipes [28] [35] [36] could address this issue.

More generally, the proposed framework generates actions that are executed in an open-loop manner. For task execution by human-robot or even robot-robot teams in the real-world, we would need to monitor the environment's and human's state and adapt the robot's actions accordingly. We find this an exciting topic for future work.

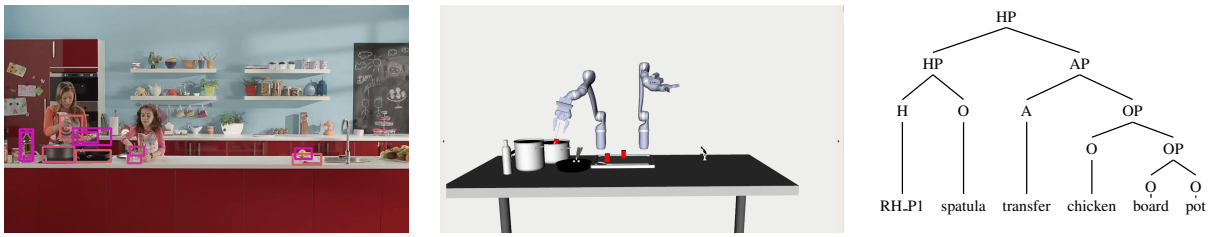
Implications. The World Wide Web contains a vast mount of online content that robots can leverage to perform tasks in human-robot and robot-robot teams. We have presented a framework that takes as input an unconstrained video with annotated object labels and outputs a human-interpretable plan. We demonstrate the execution of the plan in a simulation environment with two robotic arms and show that we can fully reproduce the actions of a simple cooking video. We find that this work brings us closer to the goal of robots executing a variety of manipulation plans by watching videos online.



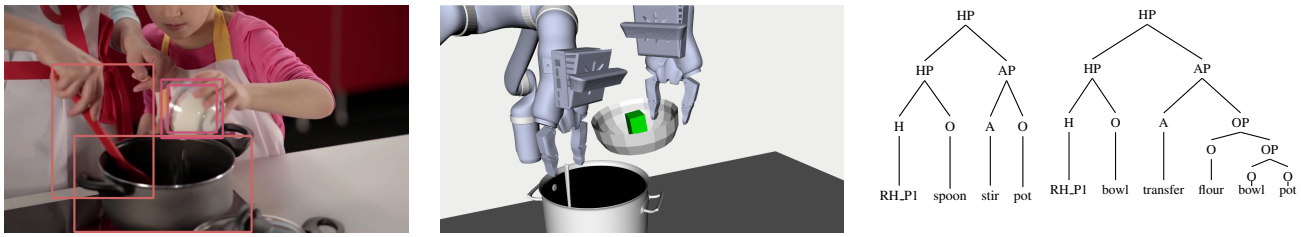
(a) The woman is handing over a lemon to the girl.



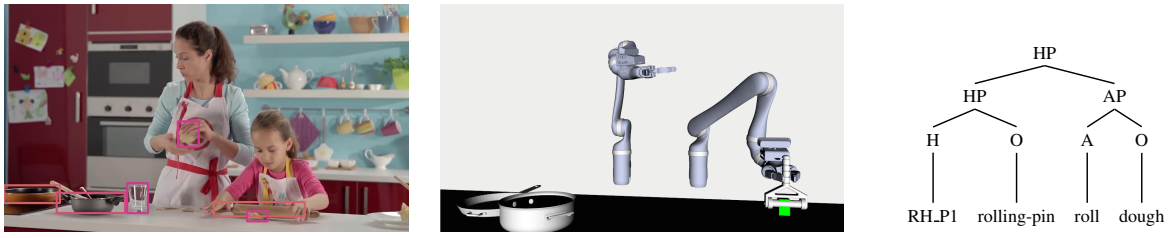
(b) The woman is holding the chopping board for the girl to cut the meat.



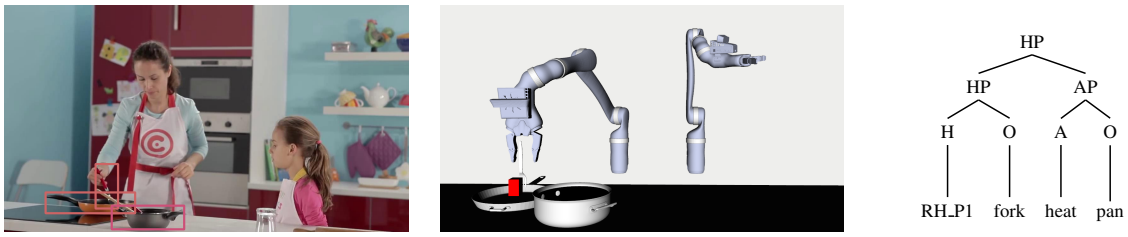
(c) The woman is transferring chicken from the chopping board to the pot.



(d) The woman is stirring the pot while the girl is transferring the flour from the bowl to the pot.



(e) The girl is rolling the dough.



(f) The woman is heating some food in the pan.

Fig. 7: Example frames, snapshots in WeCook of two robots executing the same actions with humans and generated actions trees of 6 successful cases. The captions depict the ground-truth descriptions of each successful case.

REFERENCES

[1] G. Hoffman, "Evaluating fluency in human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019.

[2] H. Zhang, P.-J. Lai, S. Paul, S. Kothawade, and S. Nikolaidis, "Learning collaborative action plans from youtube videos," in *Proceedings of the International Symposium on Robotics Research (ISRR 2019)*, Hanoi, Vietnam, 2019.

[3] Y. Yang, Y. Li, C. Fermüller, and Y. Aloimonos, "Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web." in *AAAI*, 2015, pp. 3686–3693.

[4] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[5] Y. Yang, A. Guha, C. Fermuller, and Y. Aloimonos, "A cognitive system for understanding human manipulation actions," *Advances in Cognitive Sysytems*, vol. 3, pp. 67–86, 2014.

[6] S. H. Lee, I. H. Suh, S. Calinon, and R. Johansson, "Autonomous framework for segmenting robot trajectories of manipulation task," *Autonomous robots*, vol. 38, no. 2, pp. 107–141, 2015.

[7] P. Fearnhead and Z. Liu, "Efficient bayesian analysis of multiple changepoint models with dependence across segments," *Statistics and Computing*, vol. 21, no. 2, pp. 217–229, 2011.

[8] P. A. Lasota and J. A. Shah, "Bayesian estimator for partial trajectory alignment," 2019.

[9] T. Iqbal, S. Li, C. Fourie, B. Hayes, and J. A. Shah, "Fast online segmentation of activities from partial trajectories," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5019–5025.

[10] S. Qi, B. Jia, and S.-C. Zhu, "Generalized earley parser: Bridging symbolic grammars and sequence data for future prediction," *arXiv preprint arXiv:1806.03497*, 2018.

[11] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, "Learning movement primitive libraries through probabilistic segmentation," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 879–894, 2017.

[12] D. Hallac, P. Nystrup, and S. Boyd, "Greedy gaussian segmentation of multivariate time series," *Advances in Data Analysis and Classification*, pp. 1–25, 2018.

[13] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Transactions on Circuits and Systems for Video technology*, vol. 18, no. 11, p. 1473, 2008.

[14] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[15] A. Nguyen, D. Kanoulas, L. Muratore, D. G. Caldwell, and N. G. Tsagarakis, "Translating videos to commands for robotic manipulation with deep recurrent neural networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[16] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim, "Neural program synthesis from diverse demonstration videos," in *ICML*, 2018, pp. 4797–4806.

[17] T. Shu, X. Gao, M. S. Ryoo, and S.-C. Zhu, "Learning social affordance grammar from videos: Transferring human interactions to human-robot interactions," *arXiv preprint arXiv:1703.00503*, 2017.

[18] H. Koppula and A. Saxena, "Learning spatio-temporal structure from rgb-d videos for human activity detection and anticipation," in *ICML*, 2013, pp. 792–800.

[19] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from rgb-d videos," *IJRR*, vol. 32, no. 8, pp. 951–970, 2013.

[20] H. B. Amor, G. Neumann, S. Kamthe, O. Kroemer, and J. Peters, "Interaction primitives for human-robot cooperation tasks," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 2831–2837.

[21] K. Pastra and Y. Aloimonos, "The minimalist grammar of action," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1585, pp. 103–117, 2012.

[22] N. Chomsky, *Lectures on government and binding: The Pisa lectures*. Walter de Gruyter, 1993, no. 9.

[23] D. Summers-Stay, C. L. Teo, Y. Yang, C. Fermüller, and Y. Aloimonos, "Using a minimal action grammar for activity understanding in the real world," in *IROS*. IEEE, 2012, pp. 4104–4111.

[24] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," in *arXiv preprint arXiv:1812.08008*, 2018.

[25] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph-based video segmentation," in *2010 ieee computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2141–2148.

[26] P. Jaccard, "The distribution of the flora in the alpine zone. 1," *New phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[27] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson, "One billion word benchmark for measuring progress in statistical language modeling," Google, Tech. Rep., 2013. [Online]. Available: http://arxiv.org/abs/1312.3005

[28] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, "Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[29] K. W. Church, "A stochastic parts program and noun phrase parser for unrestricted text," in *International Conference on Acoustics, Speech, and Signal Processing,*. IEEE, 1989, pp. 695–698.

[30] R. Holladay, T. Lozano-Pérez, and A. Rodriguez, "Force-and-motion constrained planning for tool use."

[31] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *Int. J. Rob. Res.*, vol. 30, no. 12, pp. 1435–1460, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1177/0278364910396389

[32] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, April 2000, pp. 995–1001 vol.2.

[33] M. Koval, P. Velagapudi, S. Choudhury, B. Hou, A. Johnson, J. King, G. Lee, J. Lee, and C. Liddick, "Aikido," https://github.com/personalrobotics/aikido.

[34] *JACO Assistive robotic arm*, 2018 (accessed November 7, 2018), https://www.kinovarobotics.com/en/products/assistive-technologies/jaco-assistive-robotic-arm.

[35] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralba, "Learning cross-modal embeddings for cooking recipes and food images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[36] C. Kiddon, G. T. Ponnuraj, L. Zettlemoyer, and Y. Choi, "Mise en place: Unsupervised interpretation of instructional recipes," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, 2015, pp. 982–992. [Online]. Available: https://www.aclweb.org/anthology/D15-1114/