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. On the other hand, previous work has shown that the space of human manipulation actions has a linguistic, hierarchical structure that relates actions to manipulated objects and tools. Building upon this theory of language for action, we propose a system for understanding and executing demonstrated action sequences from full-length, real-world cooking videos on the web. The system takes as input a new, previously unseen cooking video annotated with object labels and bounding boxes, and outputs a collaborative manipulation action plan for one or more robotic arms. We demonstrate performance of the system in a standardized dataset of 100 YouTube cooking videos, as well as in six full-length Youtube videos that include collaborative actions between two participants. We compare our system with a baseline system that consists of a state-of-the-art action detection baseline and show our system achieves higher action detection accuracy. We additionally propose an open-source platform for executing the learned plans in a simulation environment as well as with an actual robotic arm.

We build upon this theory of language for action to propose a system for zero-shot understanding both single and collaborative actions from full-length YouTube videos. Our key insight is that hands contain both spatial and temporal information of the demonstrated actions. This allows using hand trajectories to temporally segment full-length videos to short clips, derive hand-object and object-object associations and infer the demonstrated actions.

We focus on cooking tasks because of the variety in manipulation actions and their importance in home service robotics. Fig. 1 shows the components of the proposed system. We make the following contributions:

- We propose a context-free collaborative action grammar that generalizes previous work on action grammar [3] to account for collaborative actions, e.g., a person holding a plate for someone else, and object-object associations, such as “the chicken on the chopping board.” We use the grammar-based rules to derive action trees, that are more human-interpretable than deep neural network robot policies and reward functions generated by exist-
We present a visual processing system for zero-shot action understanding from full-length videos. The system detects the human hands in the video and uses the hand trajectories to split the video into clips. It then associates objects with hands and objects with other objects spatially and temporally to identify which objects are manipulated. We use the commonsense knowledge extracted from large language corpus [8], [9] to recognize individual actions. We refer to this action recognition module as a commonsense reasoning-based action recognition module. We also temporally track which person is manipulating each object, and whether the ownership of an object changes over time to identify and recognize collaborative actions.

We propose an open-source platform for executing the generated action trees in both simulation and in the real world that concatenates the action trees generated by the context-free grammar parser.

We find that the commonsense reasoning-based action recognition module works well in the cooking domain. In the FOON dataset [10], which has annotations prepared from 100 public domain YouTube cooking videos with a third-person view, the commonsense reasoning-based action recognition module achieved an average precision of 41% and recall of 69%. We additionally annotate and show the performance of the whole system in six new, previously unseen full-length YouTube videos that include collaborative actions between two persons. The precision and recall of the overall system was 0.58 and 0.51. We finally show a demonstration in simulation of two robots and in a proof-of-concept demonstration in the real world executing all the actions of one video using the open-source platform.

The current system is focused on cooking videos assigning to objects properties such as “tools”, “ingredients” and “containers.” We leave investigating its applicability to other domains for future work. An additional limitation is that the extracted action sequences are executed in an open-loop manner and thus do not withstand real-world failures or disturbances.

Nevertheless, we are excited that this work brings us a step closer to having robots generate and execute a variety of semantically meaningful plans from watching cooking videos online.

II. RELATED WORK

In this paper we propose a system 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 activity segmentation and action understanding.

Temporal Activity Segmentation. Work in activity segmentation includes learning Gaussian Mixture Models [11], specifying cost-functions incorporating spatial and temporal features of the trajectory [12] and combining classifier outputs with a symbolic grammar to parse sequence data [13]. Zhou et al. propose an end-to-end method for procedure segmentation [14]. More recently, temporal convolutional networks have been proposed to capture long-range dependencies for this task [15], [16], [17]. However these approaches rely on expensive data annotation. Thus more and more unsupervised and self-supervised learning approaches [18], [19] have been proposed to reduce the dependence on labeled data.

While our system can accommodate any state-of-the-art temporal segmentation technique [19], [16], [18], we applied the Greedy Gaussian segmentation algorithm by Hallac et al. [20]. The algorithm is applicable to general multivariate time-series data and we adopt it for segmenting videos to clips using the extracted hand trajectories, based on the insight that hands contain temporal and spatial information about manipulation actions.

Human Activity Understanding. There has been a lot of work on human activity recognition [21], [22], [23]. Recent work on deep learning approaches has enabled the generation of natural language [24], individual robot commands [25], low-level robot policies [5], [6], [7], [26] and neural programs [27] using manually annotated datasets or visual human demonstrations. Generation of collaborative actions has been achieved by representing them as social affordances [28] from data recorded in a lab setting. Pastra et
al. [29] discuss a minimalist grammar for action understanding, inspired by the suggestion by Chomsky [30]. Based on this theoretical insight, Yang et al. [3], [2] proposed a system that uses deep neural networks for hand and object detection and association, while leverages a language corpus for action recognition. Performance was shown on 12 selected clips. In our preliminary work [31], we presented a collaborative grammar whose functionality was tested only qualitatively in a small number of manually selected simple clips, rather than in full-length unconstrained videos.

Contrary to the aforementioned work, our system includes a full pipeline that takes as input full-length, previously unseen videos with annotated objects and bounding boxes, infers manipulated objects and tools based on hand detections, recognizes single and collaborative actions representing their structure using a collaborative action grammar and concatenates the actions into a temporal sequence of executable commands. Importantly, the proposed system is modular so that individual components can be replaced or combined with state-of-the-art techniques, such as CNN-based methods for action recognition [32], [33]. Moreover, different from previous work that learns to generate deep neural network robot policies [5], [6] and reward functions [7], our framework generates human-interpretable action trees since human-interpretabiliy is important for robots that work in human environments.

III. SYSTEM ARCHITECTURE

The input to the system is a full-length, previously unseen 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 system.

Hand Detection. Our work is based on the insight that hands are the main driving force of manipulation actions [2]. We use OpenPose [34], 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.

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. The trajectory of each hand is a sequence of hand positions on the image frames. We use a greedy approach [20], which formulates the segmentation of a trajectory as a covariance-regularized maximum likelihood problem of finding the segment boundaries. The result segmentation for a hand is a list of sub-sequences of hand positions. We then generate a new sequence of segments for the whole video as the union of individual segments for different hands, which we will use for action recognition. This method results in over-segmentation with some actions spanning multiple segments, which is common in segmentation algorithms [35]. Therefore, we merge segments with identical action trees in the action graph generation phase.

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 introduce a semantic hierarchy of objects, by assigning them 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 at least a pre-defined number of consecutive frames.

Ingredient-Container Association. We 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 [36] between the bounding boxes of the two objects to pair an ingredient with a container that most likely contains that ingredient.

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 object whose bounding box is the closest to the bounding box of the hand. If the closest object is an ingredient that is contained by another object, we associate the hand with its container instead.

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. We do this by associating objects with each hand with other containers or ingredients based on the Euclidean distance between the bounding boxes of the two objects. We don’t associate the grasped object with any tool object based on the assumption that a tool object will not be the receiver of any action.

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 or single 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. We recognize commonsense actions [3], using a trained language model from a recipe corpus [8].

While the current implementation of the system 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.
Given a set of candidate actions and a set of candidate objects, we extract $P(\text{Object}|\text{Action})$ for each possible bigram consisting of one object word and one action word in corpus. We then formulate action recognition task in terms of approximate sampling and prediction from the posterior distribution:

$$P(A|O_1, ..., O_n) \sim \prod_{k=1}^{n} P(O_k|A)P(A) \quad (1)$$

where $A$ is the performed action and $O_1, ..., O_n$ are the objects involved in the action respectively. We then select the most likely action.

**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.** 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. 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 also introduce an object phrase $OP$, which we use to indicate container - ingredient relationships 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 [37] to parse the constructed visual sentences [31] and output a parse tree of the specific manipulation action. The robot can then execute the action by reversely parsing the tree [3]. Fig. 5 shows the constructed trees from five different action clips.

**Action Graph Generation and Execution.** Because of over-segmentation, 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 and manipulated objects. 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 [38]: 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) [39] to specify feasible regions of target poses of the robot’s end effector in the grasp, engage and place primitives and use bidirectional rapidly-exploring random trees [40] to plan collision-free paths.

We implement the action graph as an open-source plat-
IV. EXPERIMENTS

Our first experiment evaluates the commonsense reasoning-based module for recognizing single actions in the cooking domain. Fig. 5 shows the action graph generated for the (Rita) video. The graph includes one action (stir) that was not detected by the visual processing system.

A. Action Recognition with Commonsense Reasoning

Based on the fact that in cooking people use specific tools for specific actions, such as a knife to cut and a rolling pin to roll, we hypothesize that the system will be able to recognize most actions from the objects and tools in the scene.

Dataset and Experiment Settings. We use the publicly available manipulation knowledge representation dataset called the Functional Object-Oriented Network (FOON) [10]. The dataset was prepared from 100 public domain YouTube cooking videos with a third-person view, and it includes annotated actions and objects involved in these actions. We restricted the action set to a set of common candidate actions similarly to previous work [3]. We trained our language model on the Recipe1M+ dataset [8] and the One-Billion-Words dataset [9].

Results. The average precision over all actions was 41% and the recall was 60%. Fig. 6 shows the normalized confusion matrix for our action set, with the true positive rate (recall) for each action. The action *sprinkle* and action *pour* were often misclassified as *stir*, because of the prevalence of container objects, such as bowl and pot, which skewed the inference towards the stir action. On the other hand, the performance for *cut* was high because of the presence of the *knife* tool. These results show the effectiveness of the commonsense reasoning-based module in the cooking domain.

![Normalized confusion matrix](https://github.com/icaros-usc/wecook/blob/master/figures/confusion_matrix.png)

**Fig. 6:** Normalized confusion matrix of the action recognition with commonsense reasoning. The rows indicate true values, while the columns are the predictions of the action recognition module.

**TABLE I:** Comparison of the proposed system with a baseline system regarding the precision, recall and number of correctly detected actions in the collaborative cooking videos.

|          | Ours | Ap |          | Ours | Ap |
|----------|------|----|----------|------|----|
| **Sal**  | 0.52 | 0.31 | 0.48 | 0.31 | 0.35 | 0.35 |
| **Ritch**| 0.38 | 0.22 | 0.50 | 0.22 | 9   | 4   |
| **Rita** | 1.00 | 0.30 | 0.83 | 0.50 | 5   | 3   |
| **Peixe**| 0.65 | 0.18 | 0.76 | 0.18 | 13  | 3   |
| **Massa**| 0.62 | 0.60 | 0.41 | 0.19 | 13  | 6   |
| **Rissois**| 0.68 | 0.33 | 0.43 | 0.23 | 13  | 7   |
| **Total**| 0.58 | 0.30 | 0.51 | 0.25 | 64  | 31  |

**TABLE II:** Number of Correct Trees for Different Types of Error.

|          | Total | Correct | A | HO | OO |
|----------|-------|---------|---|----|----|
| **Sal**  | 21    | 11      | 8 | 1  | 2  |
| **Ritch**| 24    | 9       | 5 | 7  | 3  |
| **Rita** | 5     | 5       | 0 | 0  | 0  |
| **Peixe**| 20    | 13      | 0 | 3  | 4  |
| **Massa**| 21    | 13      | 4 | 2  | 2  |
| **Rissois**| 19   | 13      | 3 | 1  | 2  |
| **Total**| 110   | 64      | 20| 14 | 12 |

B. Collaborative YouTube Videos

The FOON dataset does not include object bounding boxes and does not have videos with collaborative actions. We show the applicability of the entire system, shown in Fig. 2, in six public YouTube videos, where we manually did the object annotation. Although there is no existing complete system that learns the collaborative manipulation plans like ours, we construct a baseline system that consists of a state-of-the-art action detection framework [21] and compare our system with it. The results show that our system achieves significantly higher action detection accuracy.

Dataset and Experiment Settings. 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 heavily occluded. For training the baseline system, we annotated the action labels for each one second clip of the videos and human body bounding boxes for each frame, similar to the AVA dataset [41] used in the baseline method paper [21]. The videos include a total of 37030 frames and 126 executed actions of 12 different action types.

Baseline System (Ap1). We construct a baseline system based on AIA [21], a state-of-the-art end-to-end deep learning-based action detection framework. We select AIA as our baseline method because it focuses on reasoning over person-person, person-object interactions. Given the ground truth object and human body annotations, the baseline system detects actions of each person in each one second clip of the test videos using AIA. The system then aggregates the results for each video segment. The video segments are generated using the video segmentation module of the proposed system for a fair comparison. We use the pre-trained model, which is pre-trained on Kinetics-700 dataset [42] for action classification task, proposed in the AIA paper [21]. For each test
segment. We specify the precision the tree has a non-zero temporal overlap with the ground-truth ground-truth [3], and the segment corresponding to that included persons, actions and objects are identical to the We define a correct action tree when the structure and the respect to the percentage of correctly learned action trees.

5 video, we then fine-tune the pre-trained model on the other videos. Our system works better instead of executable action trees. Our system is better at detecting collaborative actions and transfer action.

Post-hoc Analysis. We perform a post-hoc analysis of the data to identify different causes of failures of our system in the generated trees. Table II shows the number of correct trees that we generated and the number of trees that were erroneous for different reasons. We identified three sources of error: 1) action recognition (A) 2) hand-object associations (HO): some objects were not visible and we did not include them in the object annotations. There were also cases where the hand would be close to an object without grasping it, resulting in error in hand-object association. 3) object-object associations (OO): similarly to the hand-object association, the bounding boxes of two objects would overlap for consecutive frames, even though the objects were not interacting.

We observe that a main source of error was hand-object associations. This occurred often when an object was not visible and we did not include it in the object annotations. For instance, in Fig. 8a the system failed to generate an action tree for the sprinkle action since the chopped onion was not annotated. A second source of error in hand-object associations was because of the distance between bounding boxes. For example, in Fig. 8c the system incorrectly infers that the girl is grasping the spoon, although the girl’s hand does not make actual contact with the spoon. Errors occurred also in the object-object associations; for example, in Fig. 8b the system incorrectly associates the oil with the dish, although the man is bringing the object back.

V. DEMONSTRATION

To demonstrate the applicability of our system, we selected the video (Rita) where our system achieved perfect precision (Table I), while it only missed recognizing one action. Fig. 7 shows the execution of the generated action trees from that video by two simulated Kinova Gen2 lightweight robotic arms in the open source platform. In the accompanying video, we show also a proof-of-concept execution of the same actions with a robot and a human, using our open-source software platform.
VI. DISCUSSION

Limitations. Our work is limited in many ways. As we observe in the post-hoc analysis section, our assumptions in how objects are associated with hands and with other objects could be limiting in the case of occlusions during collaborative actions or clutter. To deal with the occlusion, we could incorporate tracking modules for objects as we are using tracking modules for human hands. Detecting grasping types could improve the inferred associations [43]. Additionally, the YouTube videos 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 [8], [44] or using language groundings from transcripts [45] could address this issue. On the other hand, we find that explicitly reasoning over objects and human hands provides us a way to extract more information about object properties and the environment. For example, we could observe that people tend to hold the pot to stir, and they usually only handover small objects like a knife instead of a pot. The system is currently designed for cooking tasks, with objects annotated as containers and ingredients. Future work includes generalizing the system to other collaborative tasks as well, such as furniture assembly.

More generally, the proposed system generates actions that are executed in an open-loop manner. Future work will consider using the information extracted from online videos to monitor the environment and human state [46], optimize sensor placement based on the actions performed in the task [47], adapt robot actions and manage contingencies [48].

Implications. The World Wide Web contains a vast amount of online content that robots can leverage to perform tasks in human-robot and robot-robot teams. We have presented a system that takes as input a previously unseen, cooking 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 as well as in a real world environment with a human and a robotic arm 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 cooking videos online.
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] Y. Yang, A. Guha, C. Fermüller, and Y. Aloimonos, “A cognitive system for understanding human manipulation actions,” Advances in Cognitive Systems, vol. 3, pp. 67–86, 2014.

[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] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, “You only learn one representation: Unified network for multiple tasks,” arXiv preprint arXiv:2105.04206, 2021.

[5] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine, “One-shot visual imitation learning via meta-learning,” in Conference on robot learning, PMLR, 2017, pp. 357–368.

[6] T. Yu, P. Abbeel, S. Levine, and C. Finn, “One-shot hierarchical imitation learning of compound visuomotor tasks,” arXiv preprint arXiv:1810.11043, 2018.

[7] A. S. Chen, S. Nair, and C. Finn, “Learning generalizable robotic reward functions from in-the-wild human videos,” arXiv preprint arXiv:2103.16817, 2021.

[8] J. Marin, A. Biswas, F. Oli, N. Hynes, A. Salvador, Y. Aytar, I. Weber, and A. Torralba, “RecipeIm++: A dataset for learning cross-modal embeddings for cooking recipes and food images,” IEEE Trans. Pattern Anal. Mach. Intell., 2019.

[9] 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

[10] D. Paulus, A. B. Jelodar, and Y. Sun, “Functional Object-Oriented Network: Construction & Expansion,” in ICRA. IEEE, 2018.

[11] F. Sener and A. Yao, “Unsupervised learning and segmentation of complex activities from video,” in CVPR, 2018.

[12] P. A. Lasota and J. A. Shah, “Bayesian estimator for partial trajectory placement considering mobile robot trajectory,” in 2008 IEEE International Conference on Robotics and Biomimetics, 2008, pp. 6299–6308.

[13] A. S. Nikolaidis, R. Ueda, A. Hayashi, and T. Arai, “Optimal camera affordance grammar from videos: Transferring human interactions to human–robot interactions,” arXiv preprint arXiv:1703.00503, 2017.

[14] 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.

[15] N. Chomsky, Lectures on government and binding: The Pisa lectures.

[16] H. Zhang, P.-L. Lai, S. Paul, S. Kothawade, and S. Nikolaidis, “Learning collaborative action plans from youtube videos,” in ISRR 2019, Hanoi, Vietnam, 2019.

[17] L. Wang, Z. Tong, B. Ji, and G. Wu, “Tdh: Temporal difference networks for efficient action recognition,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, pp. 1895–1904.

[18] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, “Channel-wise topology refinement graph convolution for skeleton-based action recognition,” in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, pp. 13359–13368.

[19] 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.

[20] M. Grundmann, V. Kwatra, M. Han, and I. Essa, “Efficient hierarchical graph-based video segmentation,” in CVPR, IEEE, 2010, pp. 2141–2148.

[21] P. Jaccard, “The distribution of the flora in the alpine zone, 1,” New phytologist.

[22] K. W. Church, “A stochastic parts program and noun phrase parser for unrestricted text,” in International Conference on Acoustics, Speech, and Signal Processing., 1989.

[23] R. Hollanday, T. Lozano-Pérez, and A. Rodriguez, “Force-and-motion constrained planning for tool use,” Massachusetts Institute of Technology, 2019.

[24] 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.

[25] 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 ICRA. IEEE, 2018.

[26] H. Zhang, E. Heiden, S. Nikolaidis, J. J. Lim, and G. S. Sukhatme, “Auto-conditioned recurrent mixture density networks for learning generalizable robot skills,” CoRR, vol. abs/1810.00146, 2018. [Online]. Available: http://arxiv.org/abs/1810.00146

[27] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim, “Neural program synthesis from diverse demonstration videos,” in IJCL, 2018, pp. 4797–4806.

[28] 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.

[29] 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.

[30] 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 ICRA. IEEE, 2018.

[31] H. Zhang, E. Heiden, S. Nikolaidis, J. J. Lim, and G. S. Sukhatme, “Auto-conditioned recurrent mixture density networks for learning generalizable robot skills,” CoRR, vol. abs/1810.00146, 2018. [Online]. Available: http://arxiv.org/abs/1810.00146

[32] S.-H. Sun, H. Noh, S. Somasundaram, and J. Lim, “Neural program synthesis from diverse demonstration videos,” in IJCL, 2018, pp. 4797–4806.
[48] P. A. Lasota, T. Fong, J. A. Shah, et al., A survey of methods for safe human-robot interaction.