Action Recognition for Semantic Imitation Learning

Pengfei Zhao [*1] **Julen Urain De Jesus** [*1] **Georgia Chalvatzaki** [*1]

Abstract

In this work, we present action recognition approaches in the context of *Semantic Imitation Learning* (SIL). First, we explain Semantic Imitation Learning and show how action recognition in Semantic Imitation Learning can help understand and imitate human motion in robotics. Then two approaches for action recognition are presented. An object-agnostic classifier that disregards object labeling and only considers object features and a Graph Neural Network approach for action recognition. While both approaches yield good results, we can argue that both have different inductive biases. The object-agnostic classifier learns through object-characteristics instead of object labeling, while the Graph Network takes all objects in the scene and their relations into account.

1. Introduction

Humans and robots have essentially different qualities. While humans have an innate ability to reason and decision making in unseen complex environments, robots offer accuracy, speed, and repeatability. Having humans and robots complement those qualities and collaborate has been promised as a new frontier in robotics (Villani et al., 2018; Bauer et al., 2008; Mustapha et al., 2019). For example, in situations where a task is too complicated or too expensive to automate, human-robot collaborations are an ideal solution.

With humans and robots collaborating, we have to look at two challenges that need to be solved. Firstly, for the robot to anticipate and react to human actions, it needs to *understand* human activities. Secondly, it is important that humans can easily interact with the robot. Having an intuitive human-robot interface that allows the human to teach required skills by *demonstration*, facilitates the further acceptance of human-robot collaboration. This is in contrast to traditional industrial robots that require programming by specialists (Villani et al., 2018).

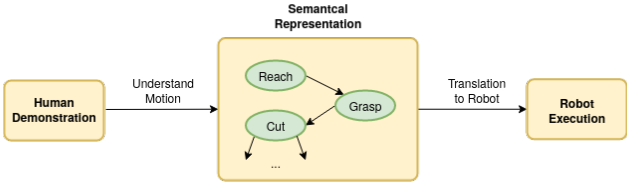

Figure 1. This figures shows an overview of how SIL could be approached. First the human motion is converted into a semantical task representation comprised of modular actions. This representation is then translated for robot execution.

Both of these challenges mentioned above are addressed with *Semantic Imitation Learning* (SIL). Unlike traditional imitation learning techniques like behavioral cloning (Torabi et al., 2018) or apprenticeship learning (Abbeel & Ng, 2004), SIL aims to abstract from trajectory-level information and represent human motion in sequences of semantic blocks. These semantic blocks describe singular actions of a broader task like *cutting* in the broader task of *making a salad*. With the robot representing human actions in a semantic space, it can translate the demonstrated motion with respect to its own embodiment. Figure 1 shows an overview of the steps in SIL. Dividing a task into modular representations of its actions has the advantage of being able to structure the task *hierarchically*. This is in accordance with the hierarchical structure of many manipulation tasks (Kroemer et al., 2019a). Another advantage is that learning modular action policies is more tractable because of a *shorter horizon* (Kroemer et al., 2019a). Moreover, modularity also offers the benefit of being able to *reuse* already learned modules in a different kind of task(Kroemer et al., 2019a).

SIL spans a wide range of sub-problems and topics. The concept map in figure 2 gives an overview of the involved topics. Here we can roughly divide concepts into two areas. One answers the question of how the robot should *understand human motion*, and the other one answers the question of how the robot should *execute the motion*.

For *human motion understanding*, we need to recognize

[*] Equal contribution [1] Department of Computer Science, Technical University Darmstadt, Germany. Correspondence to: Pengfei Zhao <pengfei.zhao@stud.tu-darmstadt.de>, Julen Urain De Jesus <urain@ias.informatik.tu-darmstadt.de>, Georgia Chalvatzaki <georgia@robot-learning.de>.

actions and objects and how they relate to each other. According to the concept of *affordance*, actions need to be understood in the context of the objects surrounding the action and certain objects *afford* certain actions. Having representations for actions and objects, we also need to structure those actions into a broader task. Here two of the most frequent approaches are logic-based or utilize graphs. Looking at the *robot execution*, we need to understand how to translate the representation of the task to the actual robot. Here different physical constraints have to be taken into account, such as the degrees of freedom of the robot or environment variables like the size of the object. Furthermore, we also need to understand when actions can be executed i.e., what are the pre-and postconditions of actions, and when actions are considered to be failed. This complements the task representation, which we have gained from before. However, policies for executing the individual action primitives still needs to be learned. This can happen for example by hardcoding a library of primitives or using reinforcement learning to learn those policies. We refer to (Karinne Ramirez-Amaro, 2019) for a detailed survey of SIL.

Overall, dealing with all these topics would go beyond the scope of this paper. With *action and object recognition* being the foundational first step of understanding and executing human motion, throughout this paper, we will set our main focus to this subfield of SIL.
Therefore the main contribution of this paper is a comparison and discussion of action and object recognition models for semantic imitation learning.

2. Related Works

In this section, we will first present works that provide a full pipeline for SIL. Next, different approaches for how to decompose a task into actions are introduced. Finally, we will focus on approaches towards action recognition.

2.1. Semantic Imitation Learning

Some works have focused on constructing full pipelines for SIL. Yang et al. (2015b) developed a system to learn manipulation tasks from constrained videos. Here CNNs are used for grasping type and object recognition. The action structure is then captured with Probabilistic Context-Free Grammars. Ramírez-Amaro et al. (2015) provided a framework suitable for *on-line* requirements. For this, the authors utilize a simple perception system based on a decision tree for action classification. The execution module then takes as input the classified action and calls motor commands that have being preprogrammed for each action. Huang et al. (2018) propose a *Conjugate Task Graph* to represent the task structure. This graph is generated with neural networks given video demonstrations together with a labeled sequence of actions. For execution, a neutral network first identifies the current node and the other network predicts the next node transition.

Finally, other approaches have been introduced that don't rely on action labeling. Therefore they circumvent the need to ground motion in language. Yu et al. (2018) decompose a task into primitive actions by predicting primitive ending with LSTMs. Policies for the segmented primitives are then learned with domain-adaptive meta-learning.

2.2. Task Decomposition

In task decomposition the goal is to describe the task through a structured representation of actions. (Konidaris et al., 2012) decompose a demonstration trajectory into a sequence of actions using changepoint detection and organize them in a tree structure.

Niekum et al. (2015) rely on Hidden Markov Models for sequencing actions while utilizing a finite-state machine to structure actions and determine state transitions with a trained classifier.

Probabilistic Context-Free Grammar has also being used to sequence tasks into hierarchical modular structures (Lioutikov et al., 2020; Yang et al., 2015b). With methods from the automated planning community, Petrick & Foster (2016) designed a action-planer that was deployed on a robotic bartender. Another frequent approach is to use graph based representations (Hayes & Scassellati, 2016; Huang et al., 2018; Neumann et al., 2009). (Neumann et al., 2009) structure a task as a graph and learn transition conditions with a SVM.

2.3. Action Recognition

The problem of recognizing actions has been a widely studied topic in both computer vision and robotics. In computer vision, CNNs have been applied to incorporate temporal and spatial information of videos for action recognition. Simonyan & Zisserman (2014) devised an architecture including two separate CNNs. The first one taking as input a single static frame, includes the spatial information. For the second CNN the input is formed by stacking optical flow fields. For prediction, the outputs of both CNNs are combined. Ji et al. (2013) instead use a single CNN performing 3D convolutions on stacked consecutive video frames. For a more detailed comparison of the use of CNNs for action recognition please see (Tran et al., 2018).

Other works build upon the Faster R-CNN framework (Gkioxari et al., 2017; Shen et al., 2018). Gkioxari et al. (2017) use Faster R-CNN to detect human and object bounding boxes. Then in a ”human-centered-branch” features are extracted from the human bounding box to predict the actions. One central idea of their approach is that human appearance is indicative of the action executed.

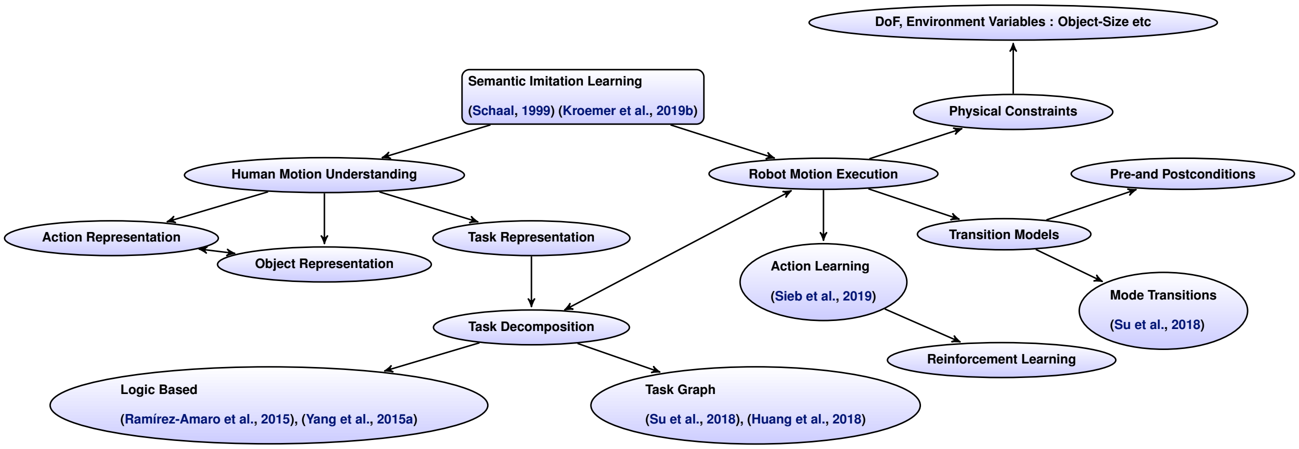

Figure 2. This map shows an overview of the different topics and fields in SIL. In general SIL involves two major subproblems. The problem of understanding and representing human motion and the problem of executing the demonstrated motion on the robot.

In previously mentioned works, object information was not explicitly used for action recognition. However, this contextual information of actions is proposed to be a key part in understanding actions (Yao et al., 2013; Kalogeiton et al., 2017). This is explained by the concept of *affordance* (Gibson, 1979). An object is said to *afford* a set actions if the properties of the object limit the possible actions performed on the object to said set. For example, a knife *affords* the action of cutting, but not drinking. There has been lots of work studying the semantical relationship between action and objects. Early works used hand-designed features and graphical models to model object affordances (Koppula et al., 2013; Kjellström et al., 2011).

More recently, object context has been included using Graph Neural Networks (Liang et al., 2020; Guo et al., 2018; Kato et al., 2018). Guo et al. (2018) devise a Few-shot 3D action recognition approach by learning a metric function that compares interaction graphs. These interaction graphs capture information about the objects and body parts involved in the scene. Kato et al. (2018) include object context using a combination between word embeddings and graphs. Here a graph is constructed from Subject-Verb-Object pairs from a knowledge base. Each action node is then linked with a verb node and noun node and stores the corresponding word embedding. Using graph convolutions, new action embeddings are learned and matched with visual features from a CNN. Dreher et al. (2020) derive a graph representation using bounding boxes for objects and hands. This graph is then processed with a Graph Neural Network to predict actions for left and right hand.

3. Methods

This section will introduce two different approaches for action recognition that are later compared with each other. We start from a simple action-object recognition approach that avoids directly learning object representations. We then proceed to an approach that includes object relation explicitly using graph representations and classifies actions with a Graph Neural Network. Both methods can be used in the context of SIL to identify modular action primitives that can be composed into tasks and executed by the robot.

3.1. Object-Agnostic Classifier

While object information is indicative of the action performed, directly including object as output prediction brings certain disadvantages. Having output labels like *cutting tomato*, where actions and objects are jointly labeled, yields a combinatorial complexity of labels. This could be circumvented with having two separated prediction branches. However, here we have the problem of not being able to handle novel objects not encountered during training time. Therefore we propose an object-agnostic architecture that predicts actions explicitly and infers object information implicitly. This is especially helpful for SIL and robotics in general since there's a vast amount of different objects that a robot might need to handle, and having labels for all of them is prohibitive.

3.1.1. Architecture

For the object-agnostic classifier, we have designed the architecture as shown in Figure 3. The key idea is that we only use the object's characteristics for prediction, but do not ground the object in semantic labels. Therefore even when a novel object is encountered, we do not need to

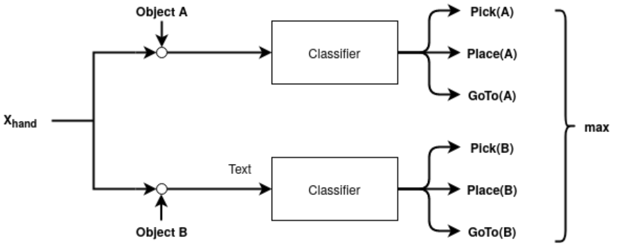

Figure 3. This figure shows the configuration of the object-agnostic action recognition model during prediction time. The same classifier is used for different combinations of hand and object trajectories as input.

have a label for it. Instead, we concatenate the motion data for object and hand and use the concatenated data as the classifier's input. During training, the classifier will learn to predict the correct action when the correct object is co-occurring. Here it is crucial also to learn when an object is not involved by predicting the *no-action* class. We make use of the same classifier for all objects. For example, if we want to learn the action-object relation *pick tomato*, then hand and tomato motion data are concatenated. Given this input, the classifier is tasked to learn *pick*. For all other non-involved objects in the scene, the same concatenation is performed with the hand motion. The difference is that the *no-action* state should be predicted. We can infer the object involved by comparing the classifier output with outputs for other objects during prediction time. This way, there is no need to include objects labels. If we encounter novel objects similar to the objects seeing during training, we can distinguish them through comparison.

For the classifier, we used a multilayer perceptron with one hidden layer and ReLU activations. The hidden layer comprises of 100 neurons. Since one hidden layer already yielded good results on the validation set, experiments with multiple hidden layers were not conducted. The amount of neurons was chosen, since additional neurons did not improve performance on the validation set as can be seen in Figure 4. We can see that the cross-entropy loss for an architecture with 100 neurons converge to a similar validation set performance as compared to an architecture with 130 neurons. We used a non-overlapping sliding window consisting of 5 samples to extract input data from the motion trajectories. The loss function is the cross-entropy between the predicted action distribution and the ground truth action distribution. We train for 3 epochs with a learning rate of 0.001.

3.1.2. Dataset

We have used a motion capturing system to record position and orientation trajectories for hand and various objects.

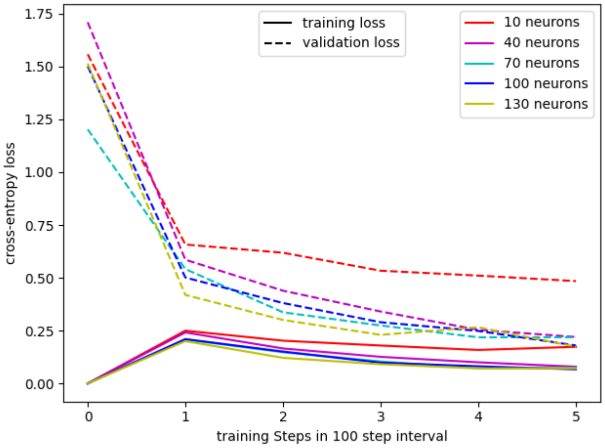

Figure 4. The cross-entropy loss is plotted against the amount of training steps measured in intervals of 1000. We can see that the validations set performance for 100 and 130 neurons converge towards similar results.

In detail, the dataset consists of time samples of position and orientation for the action-object combinations : *go-to-glass, go-to-tomato, take-glass-to-dish, go-to-tea, pick-tomato, leave-dish, pick-glass, go-to-dish* and *pick-dish*. For each of these actions, 7 state variable were recorded. The x, y and z position as well as a 4-dimensional unit quaternion vector describing the orientation. On average each action was executed 20 times. In total we have 6334 time samples for each state variable. Examples of the motion data for *go-to-glass* can be seen in Figure 5.

3.2. Graph Network Classifier

A more holistic approach to including object information is representing the object relations in the scene with a graph structure. Dreher et al. (2020) devised a pipeline for constructing graphs from RGB-D videos and using a Graph Neural Network to simultaneously classify actions for the left and right hand.

The pipeline follows the following steps :

- Identify bounding boxes for objects and left and right hand in the scene
- Determine object relations
- Construct a graph given objects and relations
- Classify actions using the Graph Neural Network

3.2.1. Graph Construction

The input of the pipeline are consecutive RGB-D images. In the first step 2D object boundary boxes are computed with

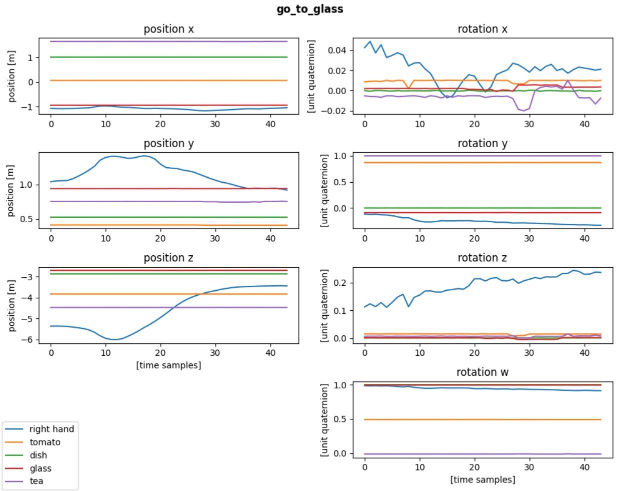

Figure 5. Here exemplary position and orientation trajectories for the action *go-to-glass* is shown. On the left side, the lines show position for each of the five objects including the hand motion. On the right side, the same is depicted for the orientation given in unit quaternions.

YOLO (Redmon & Farhadi, 2018). For the hand bounding boxes, OpenPose (Cao et al., 2019) is used. They then combine the depth information to construct 3D bounding boxes. Finally, the spatial relations between objects are determined from the set of possible relations: *contact, above, below, left, right, front, behind, inside, surround, moving together, halting together, fixed moving together, getting close, moving apart, stable* . The graphs are constructed per frame and are later concatenated along a new temporal dimension before inputting to the Graph Network.

3.2.2. Graph Classification

For classification Dreher et al. (2020) utilize a Graph Neural Network. Graph Neural Networks, as introduced in (Battaglia et al., 2018), are designed to operate on graph like structures. Following definitions from (Battaglia et al., 2018), a graph is given by a 3-tuple $G = (u, V, E)$. Here u is a global attribute of the graph, V and E the set of nodes and edges respectively. For each $v_a \in V$ is represented by a node attribute. Each $e \in E$ is given by $e = (e_a, s, r)$, where e_a is the edge attribute, s and r represent sender and receiver nodes. A Graph Network takes this graph as input and process its attributes and return the updated graph. A Graph Network can come in different types and can also be composed into bigger Graph Networks from smaller ones.

For the classification the *encode-process-decode* architecture was used (Battaglia et al., 2018). Here there are three components: *encoder*, *decoder* and *core*. Each of the components is represented by a Graph Network. This model takes the constructed scene graph G_{in} as input, where edge attribute e_a is given by the object relation and node attribute v_a is the detected object class. The output of the classifier is a probability distribution over possible actions encoded by the global attribute u. To also train the classifier for the left hand, the constructed scene graph is simply mirrored.

Table 1. Performance metric for action-only prediction

Action	Precision	Recall	F1
NO-ACTION	0.99	0.96	0.97
GO-TO	0.78	0.89	0.83
PICK	1.0	0.93	0.96
LEAVE	0.78	0.96	0.86

3.2.3. Dataset

The dataset used is given by (Dreher et al., 2020). The dataset consists of RGB-D video data showing different bimanual action sequences. In total 6 subjects were recorded doing 14 different actions : *idle, approach, retreat, lift, place, hold, stir, pour, cut, drink, wipe, hammer, saw, screw*. Furthermore the subjects interacted with 12 different objects : *up,bowl, whisk, bottle, banana, cutting board, knife, sponge, hammer, saw, wood, screwdriver*. In total the dataset comprises of 540 recordings with a runtime of 2h 18min.

4. Experiments

4.1. Object-Agnostic Classifier

We tested the object-agnostic classifier on a dataset containing object and hand motion trajectories. Trajectories were recorded for the objects: *dish, glass, tea, tomato*. In addition following actions were executed : *go-to, leave, pick, take*. The trajectories consist of position and orientation samples.

The classifier was tested on predicting action alone and prediction action and object combinations. The training was done on 60% of the data, and testing with 30% of the data. The remaining data were used for validation.

Action Only Prediction We first evaluate how the classifier performs when we only predict the actions. Actions are predicted given a time slice of 5 samples. Results can be seen in Table 1. For the confusion matrix please see Figure 6.

Action-Object Prediction We evaluate the performance of predicting action explicitly and objects implicitly according to architecture illustrated in Figure 3. As before, we take time slices of 5 samples as input for the classifier. The performance metrics can be viewed in Table 2 and the confusion matrix in Figure 7.

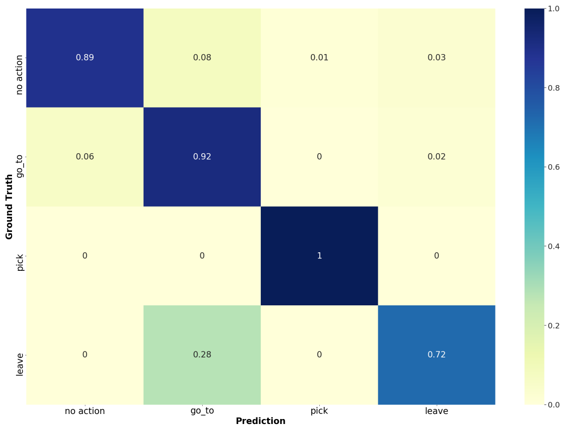

Figure 6. Classification performance for action-only prediction shown as a row-normalized confusion matrix.

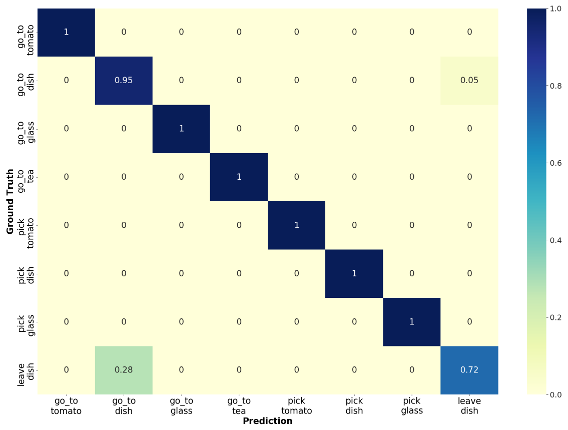

Figure 7. Classification performance for action-object prediction shown as a row-normalized confusion matrix.

Table 2. Performance metric for action-object prediction

ACTION	PRECISION	RECALL	F1
GO-TO TOMATO	1.0	1.0	1.0
GO-TO DISH	0.94	0.78	0.86
GO-TO GLASS	1.0	0.93	0.96
PICK TOMATO	1.0	1.0	1.0
PICK DISH	1.0	1.0	1.0
PICK GLASS	1.0	1.0	1.0
LEAVE DISH	0.72	0.92	0.81

Table 3. Performance metric for Graph Network Classifier

ACTION	PRECISION	RECALL	F1
IDLE	0.85	0.71	0.78
APPROACH	0.31	0.41	0.35
RETREAT	0.34	0.43	0.38
LIFT	0.32	0.50	0.39
PLACE	0.34	0.45	0.38
HOLD	0.82	0.64	0.72
POUR	0.66	0.65	0.66
CUT	0.74	0.67	0.70
HAMMER	0.64	0.56	0.60
SAW	0.68	0.58	0.63
STIR	0.92	0.84	0.88
SCREW	0.76	0.79	0.77
DRINK	0.70	0.71	0.70
WIPE	0.78	0.87	0.82

We can observe from Figure 6 that most of the misclassifications can be attributed to confusing *go-to* with *no action* and *go-to* with *leave*. In the first case, this could be possibly explained by the reason that *go-to* mostly involves hand motion whereas objects remain still. This is similar to when the classifier predicts *no action*. That is when object trajectory and hand motion trajectory taken as input are not linked to an action. For the confusion between *go-to* and *leave*, one possible explanation might be a disproportionate dataset. The action *leave* was only executed in combination with one object, whereas for action *go-to* 4 action-object combinations exist. Furthermore, through the comparison between the confusion matrices 6 and 7, we can see that there are actually fewer misclassifications between actions when predicting actions and objects. This is surprising, since we would expect lower performance due to an additional prediction task. However, looking more closely, we actually include additional information through a comparison of the classifier output for different objects. In other words, even-tough the hand motion trajectories remain the same, having different object trajectories as additional input changes the predicted action. Looking at this result, we could conclude that including object information that are not directly involved with the action could benefit our action recognition. This could also be a hint at a possible benefit of using graph structures to capture object information in the scene.

4.2. Graph Network Classifier

Results for the Graph Network are based on (Dreher et al., 2020). For evaluation a leave-one-subject-out cross-validation is performed to obtain 6 folds of training and testing sets. The performance metrics are listed in Figure 3. Furthermore, Figure 8 shows the confusion matrix. The authors argue that some actions could not be distinguished because the scene graph needs to be extended with additional information. For example in order to distinguish between drinking and holding, one needs to add the head to the scene graph. Another problem encountered is the detection of bounding boxes for very thin objects like hammers.

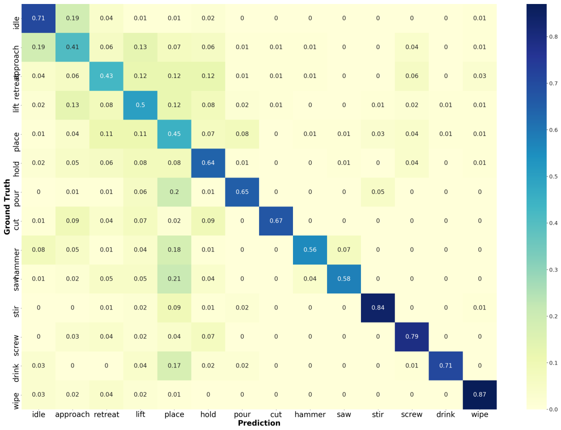

Figure 8. Classification performance for top action performance shown as normalized confusion matrix.

5. Conclusions

The previously studied two models follow two different approaches. While the Graph Network represents object relations with a graph structure, the object-agnostic classifier avoids object labeling and only learns to predict actions from the object characteristics. We can say that the Graph Network has a different *inductive bias* than the object-agnostic classifier. An *inductive bias* is a structural difference of the learning algorithm causing it to prefer certain solutions or interpretations over others. For CNNs, the bias would be to prefer locality and translational invariance.

In our case, both *inductive biases* for the Graph Network and the object-agnostic classifier have meaningful interpretations for SIL learning and robotics. On the one hand, the graph structure captures a holistic scene understanding involving object relations. This is helpful for SIL, since the actions the robot needs to learn have to be understood in the context of the surrounding objects. On the other hand, for the object-agnostic classifier learning the meaning of an object for a particular action through the object's characteristics instead of its labeling has an inherent benefit. For example, *orange* and *tomato* have different labeling, but a similar meaning for the action *cutting*.

Therefore in future work, we hope to combine the benefits of *inductive biases* from both models. We would like to use learned feature representations of objects characteristic as object node attributes for the Graph Neural Network. This is a modification to the Graph Neural Network from (Dreher et al., 2020), who uses object labels as node representations.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, pp. 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V. F., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gülçehre, Ç., Song, H. F., Ballard, A. J., Gilmer, J., Dahl, G. E., Vaswani, A., Allen, K. R., Nash, C., Langston, V., Dyer, C., Heess, N., Wierstra, D., Kohli, P., Botvinick, M., Vinyals, O., Li, Y., and Pascanu, R. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018. URL http://arxiv.org/abs/1806.01261.

Bauer, A., Wollherr, D., and Buss, M. Human-robot collaboration: a survey. *I. J. Humanoid Robotics*, 5:47–66, 03 2008. doi: 10.1142/S0219843608001303.

Cao, Z., Martinez, G. H., Simon, T., Wei, S.-E., and Sheikh, Y. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Dreher, C. R. G., Wächter, M., and Asfour, T. Learning object-action relations from bimanual human demonstration using graph networks. *IEEE Robotics and Automation Letters*, 5:187–194, 2020.

Gibson, J. J. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Gkioxari, G., Girshick, R. B., Dollár, P., and He, K. Detecting and recognizing human-object interactions. *CoRR*, abs/1704.07333, 2017. URL http://arxiv.org/abs/1704.07333.

Guo, M., Chou, E., Huang, D.-A., Song, S., Yeung, S., and Fei-Fei, L. Neural graph matching networks for fewshot 3d action recognition. In *ECCV*, 2018.

Hayes, B. and Scassellati, B. Autonomously constructing hierarchical task networks for planning and human-robot collaboration. pp. 5469–5476, 05 2016. doi: 10.1109/ICRA.2016.7487760.

Huang, D., Nair, S., Xu, D., Zhu, Y., Garg, A., Fei-Fei, L., Savarese, S., and Niebles, J. C. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. *CoRR*, abs/1807.03480, 2018. URL http://arxiv.org/abs/1807.03480.

Ji, S., Xu, W., Yang, M., and Yu, K. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35: 221–231, 2013.

Kalogeiton, V., Weinzaepfel, P., Ferrari, V., and Schmid, C. Joint learning of object and action detectors. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2001–2010, 2017.

Karinne Ramirez-Amaro, Yezhou Yang, G. C. A survey on semantic-based methods for the understanding of human movements. *Robotics and Autonomous Systems*, 119: 117–122, 2019.

Kato, K., Li, Y., and Gupta, A. Compositional learning for human object interaction. In *ECCV*, 2018.

Kjellström, H., Romero, J., and Kragic, D. Visual object-action recognition: Inferring object affordances from human demonstration. *Comput. Vis. Image Underst.*, 115: 81–90, 2011.

Konidaris, G., Kuindersma, S., Grupen, R., and Barto, A. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31:360 – 375, 2012.

Koppula, H., Gupta, R., and Saxena, A. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 32:951 – 970, 2013.

Kroemer, O., Niekum, S., and Konidaris, G. D. A review of robot learning for manipulation: Challenges, representations, and algorithms. *CoRR*, abs/1907.03146, 2019a. URL http://arxiv.org/abs/1907.03146.

Kroemer, O., Niekum, S., and Konidaris, G. D. A review of robot learning for manipulation: Challenges, representations, and algorithms. *CoRR*, abs/1907.03146, 2019b. URL http://arxiv.org/abs/1907.03146.

Liang, Z., Guan, Y., and Rojas, J. Visual-semantic graph attention network for human-object interaction detection. *ArXiv*, abs/2001.02302, 2020.

Lioutikov, R., Maeda, G., Veiga, F., Kersting, K., and Peters, J. Learning attribute grammars for movement primitive sequencing. *The International Journal of Robotics Research*, 39(1):21–38, 2020. doi: 10.1177/0278364919868279. URL https://doi.org/10.1177/0278364919868279.

Mustapha, A., Maoudj, A., Isma, A., and Hentout, A. Human-robot interaction in industrial collaborative robotics: a literature review of the decade 2008-2017. *Advanced Robotics*, pp. 1–36, 07 2019. doi: 10.1080/01691864.2019.1636714.

Neumann, G., Maass, W., and Peters, J. Learning complex motions by sequencing simpler motion templates. In *ICML '09*, 2009.

Niekum, S., Osentoski, S., Konidaris, G., Chitta, S., Marthi, B., and Barto, A. G. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015. doi: 10.1177/0278364914554471. URL https://doi.org/10.1177/0278364914554471.

Petrick, R. P. A. and Foster, M. E. Using general-purpose planning for action selection in human-robot interaction. In *AAAI Fall Symposia*, 2016.

Ramírez-Amaro, K., Beetz, M., and Cheng, G. Understanding the intention of human activities through semantic perception: Observation, understanding and execution on a humanoid robot. *Advanced Robotics*, 29:345–362, 03 2015. doi: 10.1080/01691864.2014.1003096.

Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018.

Schaal, S. Is imitation learning the route to humanoid robots? 3(6):233–242, 1999. URL http://www-clmc.usc.edu/publications/S/schaal-TICS1999.pdf;http://www-clmc.usc.edu/publications/S/schaal-TICS1999-rep.pdf. clmc.

Shen, L., Yeung, S., Hoffman, J., Mori, G., and Fei-Fei, L. Scaling human-object interaction recognition through zero-shot learning. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1568–1576, 2018.

Sieb, M., Zhou, X., Huang, A., Kroemer, O., and Fragkiadaki, K. Graph-structured visual imitation. *CoRR*, abs/1907.05518, 2019. URL http://arxiv.org/abs/1907.05518.

Simonyan, K. and Zisserman, A. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

Su, Z., Kroemer, O., Loeb, G., Sukhatme, G., and Schaal, S. Learning manipulation graphs from demonstrations using multimodal sensory signals. In *IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia, May 21–25 2018.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. *CoRR*, abs/1805.01954, 2018. URL http://arxiv.org/abs/1805.01954.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. A closer look at spatiotemporal convolutions for action recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6450–6459, 2018.

Villani, V., Pini, F., Leali, F., and Secchi, C. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, 03 2018. doi: 10.1016/j.mechatronics.2018.02.009.

Yang, Y., Aloimonos, Y., Fermüller, C., and Aksoy, E. E. Learning the semantics of manipulation action. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 676–686, Beijing, China, July 2015a. Association for Computational Linguistics. doi: 10.3115/v1/P15-1066. URL https://www.aclweb.org/anthology/P15-1066.

Yang, Y., Li, Y., Fermüller, C., and Aloimonos, Y. Robot learning manipulation action plans by " watching " unconstrained videos from the world wide web. 01 2015b.

Yao, B., Ma, J., and Fei-Fei, L. Discovering object functionality. *2013 IEEE International Conference on Computer Vision*, pp. 2512–2519, 2013.

Yu, T., Abbeel, P., Levine, S., and Finn, C. One-shot hierarchical imitation learning of compound visuomotor tasks. *ArXiv*, abs/1810.11043, 2018.