2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

The Robot as Scientist: Using Mental Simulation to Test Causal Hypotheses Extracted from Human Activities in Virtual Reality

Constantin Uhde[1], Nicolas Berberich[1], Karinne Ramirez-Amaro[2] and Gordon Cheng[1]

***Abstract*— To act effectively in its environment, a cognitive robot needs to understand the causal dependencies of all intermediate actions leading up to its goal. For example, the system has to infer that it is instrumental to open a cupboard door before trying to grasp an object inside the cupboard. In this paper, we introduce a novel learning method for extracting instrumental dependencies by following the scientific approach of observations, generation of causal hypotheses, and testing through experiments. Our method uses a virtual reality dataset containing observations from human activities to generate hypotheses about causal dependencies between actions. It detects pairs of actions with a high temporal co-occurrence and verifies if one action is instrumental in executing the other action through mental simulation in a virtual reality environment which represents the system's mental model. Our system is able to extract all present instrumental action dependencies while significantly reducing the search space for mental simulation, resulting in a 6-fold reduction in computational time.**

I. INTRODUCTION

Research in cognitive science suggests that children learn causal hypotheses through the combination of observation and testing [1], resembling the scientific approach and thus giving evidence to *"The Child as Scientist"* theory. This constructivist epistemology, which can be traced back to Jean Piaget's theory of cognitive development, requires the existence of mental models of how one's environment works, which can be dynamically constructed and tested through mental simulation. Similarly, we propose *"The Robot as Scientist"* approach according to which a cognitive system observes human activities, generates causal hypotheses about instrumental actions from these observations, and then tests these hypotheses either in the physical environment or through an appropriate mental simulation of it. This is based on the approach of purposive learning for robots as discussed in [2].

In contrast to natural language processing, where sentences can be interpreted in the presence of word sequence errors, in the robotics domain a wrong sequence of actions in action planning propagates and can render subsequent actions futile. In everyday human activities, some actions are instrumental for the execution of subsequent actions, e.g. opening a cupboard door is an instrumental action for subsequently grasping a plate inside the cupboard (see Fig. 1) and turning on the water tap is instrumental for cleaning a plate in the sink. Therefore, machine learning sequence-to-sequence models for purposive action sequence generation need to be

[1]Institute for Cognitive Systems, Technical University of Munich
{c.uhde, n.berberich, gordon}@tum.de
[2]Chalmers University of Technology, Gothenburg
karinne@chalmers.se

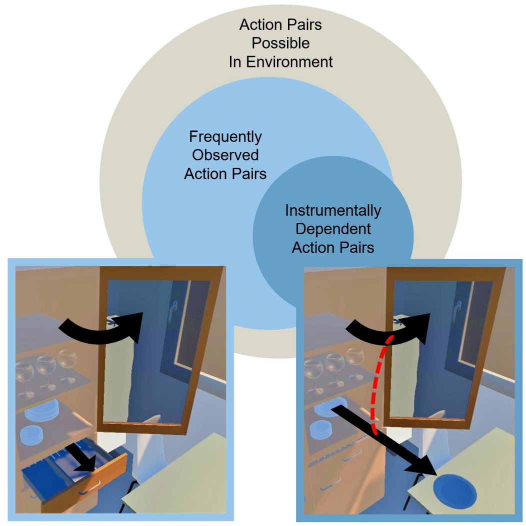


Fig. 1: Instead of testing all action pairs which are possible in the environment for their instrumental dependency on each other, it is more efficient to first observe human activities and only consider those action pairs which co-occur in the right temporal sequence as candidates for causal dependency to be tested through physical or mental experiments.

augmented with semantic knowledge about strictly required action dependencies. Relying on classical good-old-fashioned AI (GOFAI) [3], where these action sequence rules would be predefined by humans as so-called production rules, can lead to the well-known problems of not being able to adapt to dynamic and unforeseen changes in environments and not adapting to the cognitive system's individual embodiment and functionality. Instead, a hybrid combination of data-driven learning and mechanistic rule testing and construction is required which is proposed in this paper. As illustrated in Fig. 2, this is achieved by recording an observational dataset of human activities and extracting pairs of actions from it where one action (prior action) frequently occurs before the second action (posterior action) occurs. These action pairs are considered causal hypotheses for instrumental dependencies and are tested through executability checks in the virtual environment (mental simulation). While supervised machine learning is a behaviorist approach for finding associations between input and output variables (e.g. stimuli and responses) and symbolic AI is based on cognitivism, our hybrid approach can be considered a step towards constructivist AI in which data-driven and mechanistic methods are synergistically combined.

II. RELATED WORK

Using human activity observation for mental simulation has been a topic related to imitation learning for some time.

When discussing learning from human activities, roboticists usually refer to curated datasets based on expert demonstrations [4]. As a result, most related work is based on the assumption that the dataset only contains information that is meaningful towards the completion of the task. There have been several publications covering learning from such demonstrations at the trajectory level [5][6] or action level [7][8]. One very important element to consider when learning new tasks are the mental simulation and physical rehearsal capabilities [9]. This mental simulation allows the learning system to explore different possibilities which leads to better-learned policies even when different dynamics are explored. For example, [10] proposed a learning method that mapped the observed action with the learned primitive. The authors proposed to use physical rehearsal and mental simulation to learn as many possibilities to improve the learned policies in very dynamic situations. To develop a mental simulation framework, Virtual Reality (VR) systems are the best-suited option since they allow to collect realistic information of the demonstrations in a structured manner [11], [7]. Recent approaches are using VR to bootstrap the learning of human or robot actions using semantic-based approaches [12], [7]. Semantic-based approaches map the continuous real-world signals into meaningful symbolic descriptions. Such approaches are often considered as hierarchical methods, where different levels of abstractions are obtained and analyzed. For example, [13] presented a hierarchical method that extracted a set of grammars from human activities. Aksoy et al. [8] introduced the approach named Semantic Event Chain which is based on the affordance principle. In contrast to the robot scientist theory developed by King et al. [14], our system is not built to answer scientific questions in fields such as genetics. However, similar to children, it utilizes the scientific approach of observations, causal hypotheses, and experimentation to understand how to effectively act in its current environment. It has been shown that causal structures can be inferred through interventions and controlled experiments [15][16].

III. METHODS

A. Observing Human Activities & Action Recognition

The data being used in this work has been recorded in a virtual reality (VR) setup. Consumer VR hardware combined with well-supported 3D engines like *Unity3D* provides a good basis to quickly create and distribute new scenarios for recording human activities. Additionally, being able to fully access the environment state makes recognizing actions easier compared to real-world datasets like the *TUM Kitchen Data Set* [17] where one has to employ computer vision methods on video recordings to interpret the environment and extract the observed actions. Our dataset named *Household Activities from Virtual Environments (HAVE)* [18] consists of three scenarios, each depicting a different household task: *Setting a Table*, *Washing Dishes* and *Cleaning a Living Room*. The environments were set up to allow for variation in the sequence of actions within the given task, to gather data that resembles real-world variance. The dataset has been

TABLE I: Dataset distribution of recordings

Goal	Recordings
Setting a Table	83
Washing Dishes	96
Cleaning a Living Room	61
Total	240

recorded at the *Automatica*[1] *Trade Fair 2018* in Munich with the recording setup shown in Fig. 3. It consists of 240 human activities performed by trade fair visitors across the three scenarios. Visitors were allowed one recording per scenario. Each recording is limited to a maximum of 5 minutes and all participants were new to the scenarios. The participants had a brief adaptation phase of several seconds before they were given the scenario-specific activity goal. The recordings were done over four days, with three *HTC Vive* systems set up simultaneously. Each scenario is designed inside a 2 by 2-meter square to fit the three recording systems in the same booth and to avoid the need for virtual locomotion which would introduce trajectory discontinuities. For this work, *Setting a Table* was chosen since the importance of identifying instrumental actions such as opening the cupboard doors and drawers is evident.

The dataset is provided online[2] and consists of a third-person video, a first-person video, and CSV table of object trajectories.

In the scope of this work, we focus on the analysis of two action types which are relevant for *Setting a Table*: “Put <object> on table” and “Open/close door/drawer”. The automatic action detection is being grounded using grasp events such as grasping and releasing. These binary grasp events are directly recorded when the main VR controller button is pressed and released, together with object and hand trajectories. Based on this information, segments of *grasp-translate-release* are being grouped together. When the translated object has the predefined affordance "open-close" such as the cupboard doors and drawers, the travel direction along the single degree of freedom determines whether the action is recognized as “opening” or “closing”. For objects with the affordance “put on table”, the spatial relation to the table is measured when it is released. If the release happens directly above the table, a "Put on Table" action is recognized. These detected labels and their corresponding objects are stored for every recording run in a temporal order.

B. Generating Causal Hypotheses of Instrumental Actions

The resulting library of recorded action sequences is analyzed for temporal action co-occurrences. Algorithm 1 describes the process. First, all actions which are possible in the environment, are stored in $\mathcal{A}$. From this, for each possible action pair $\langle a_i, a_j \rangle$ the recordings are searched for cases where a_i comes before a_j. These incidents are counted for all action pairs, stored in a temporal co-occurrence matrix C and normalized by the total number of occurrences of the posterior action a_j. Effectively, this gives a look-up value for each pair, where $C(i, j)$ describes how commonly a_i

[1]https://automatica-munich.com
[2]https://github.com/TUM-ICS/HAVE-Dataset

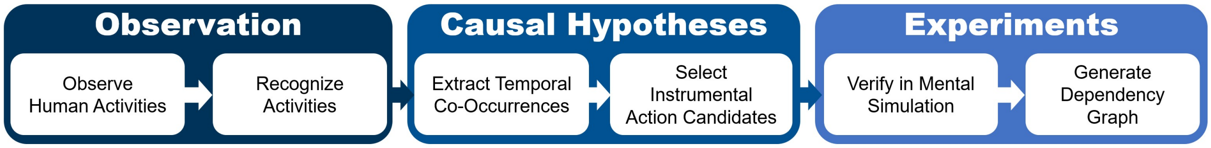
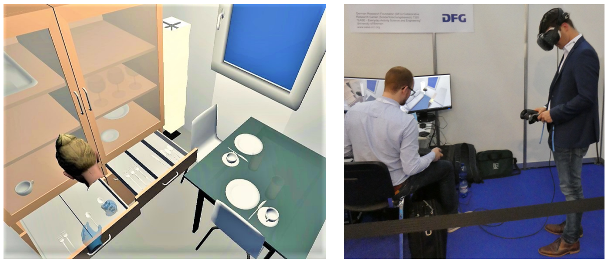


Fig. 2: The steps for extracting the action dependencies from human activities. Each step can be attributed to one of the three parts of the scientific approach. Observations in Fig. 3, causal hypotheses (Algorithm 1), experiments in Fig. 4 and the resulting dependency graph in Fig. 6. A video demonstration of the procedure can be seen under the following link: https://youtu.be/eVThEsepfbw



(a) The *Setting a Table* scenario mid activity

(b) Recording an activity at *Automatica*

Fig. 3: The recording setup in the *Automatica Trade Fair 2018* booth used for collecting human activity data.

occurs before a_j. Due to the causal nature of instrumental dependencies, where the temporal sequence is important and one action has to come before the other, the resulting matrix is asymmetric, in contrast to a correlation matrix. An example for such a co-occurrence matrix C can be seen in Fig. 5. Our action recognition system may misclassify sequences in the recording, therefore it is possible that pairs of dependent actions are not registered as co-occurring consistently. If action a_i (e.g. opening the right cupboard door) is misclassified as action a_k (e.g. opening the right drawer) then there will be cases in which plate 1 is put on the table without the system having detected opening the cupboard door beforehand, resulting in $C(i, j)$ being less than 1.0. Due to this immanent uncertainty, we use thresholding to find candidates for dependent action pairs. We mark every entry $C(i, j)$ which is below the chosen threshold τ with a 0, meaning that action i is not instrumental for action j. The candidates of action pairs in which one action frequently occurred before the other action and thus $C(i, j) \geq \tau$ are causal hypotheses of instrumental dependencies which the system needs to test through experiments.

C. Experiments in Mental Simulation

To identify which prior-posterior action pairs in our candidates list contain a dependency, we use mental simulation. We use the same virtual environment in which the dataset was recorded as a mental model for the system in which it can simulate counterfactual scenarios. For example, the system can check for different environment states whether certain actions such as *open right cupboard door* can be executed successfully. In our kitchen scenario, actions are considered to be executed successfully if their target state can be reached without collision. For opening and closing actions, this is checked by moving the object in the simulation along its single degree of freedom towards the target state while checking for collisions with other objects. For placing actions we test through a planning algorithm whether there is at least

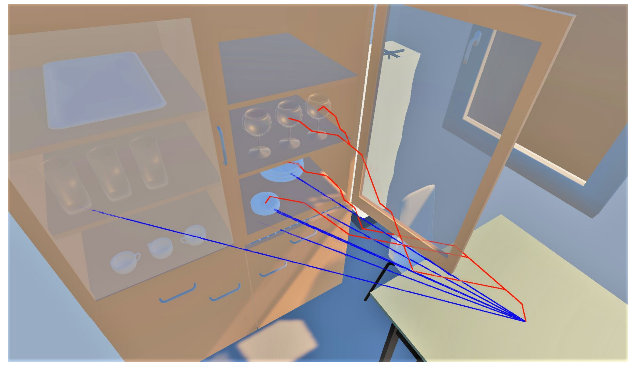


Fig. 4: The mental simulation model running in the *Setting a Table* scenario. Red lines represent valid paths while blue lines represent blocked paths. After opening the right cupboard door (instrumental action), the objects behind it can be successfully put on the table. The exception is plates which are below other plates.

one possible path between the object's initial position and its goal (Fig. 4). For the latter, an A* planner is used [19]. Since instrumental actions are required for the execution of their dependent action, we evaluate two cases for each prior-posterior candidate pair $C(i, j) \geq t$. We check if the posterior action a_j is possible without action a_i being executed prior to it and in the second case if a_j is possible after performing a_i. If a_j is only possible in the latter case, we conclude that action a_i is an instrumental action for action a_j.

$$\begin{array}{ll} & a_j \text{ is not possible without } a_i \text{ executed before} \\ & a_j \text{ is possible with } a_i \text{ executed before} \\ \hline \therefore & a_i \text{ is instrumental for } a_j \end{array}$$

This information is stored in a dependency matrix $\mathcal{D}(i, j)$. If a_j is possible in both cases, there exists no dependency and if both cases fail, the dependency of a_j can not be resolved yet. After the first pass over all pairs, multi-level dependencies are still possible and not resolved in $\mathcal{D}$, e.g. the dependency of the posterior action “Put Plate 2 on Table” on the prior action ”Open Right Door”. However, the system has recognized that putting plate 1 on the table is dependent on opening the right door and that putting plate 1 on the table is itself a candidate for being instrumental for putting plate 2 on the table. Therefore, our system chains both dependency pairs together and executes both “Open Right Door” and “Put Plate 1 on Table” (two prior actions) and then tests “Put Plate 2 on Table”. Since this results in successful execution, the system reasons that “Put Plate 2 on Table” is a 2nd-level dependency of “Open Right Door” (see Fig. 6). Other 2nd-level and 3rd-level dependencies are detected analogously.

IV. RESULTS

After feeding the 83 recordings of human activities from the *Setting a Table* scenario to our temporal co-occurrence

Algorithm 1 Temporal Co-Occurrence

Input: $O(t)$: observation trials of action sequences
$\mathcal{A}$: list of performed actions
n: number of actions
$C \leftarrow$ initialize $n \times n$ co-occurrence matrix with zeros
$f \leftarrow$ initialize action frequency vector of length n with zeros
for all observation trials $O(t_k)$ **do**
  for all ordered pairs $\langle a_i, a_j \rangle$ of actions in $\mathcal{A}$ **do**
    if action a_j occurs in observation trial $O(t_k)$ **then**
      $f(j) \leftarrow f(j) + 1$
    end if
    $p_j \leftarrow$ index of first occurrence of a_j in $O(t_k)$
    $p_i \leftarrow$ index of first occurrence of a_i in $O(t_k)$
    if $p_i \leq p_j$ **then**
      $C(i, j) \leftarrow C(i, j) + 1$
    end if
  end for
end for
for all entries $C(i, j)$ in C **do**
  $C(i, j) \leftarrow C(i, j)/f(j)$
end for
return C

Algorithm 1 we obtain a matrix C of 39×39 elements with rows representing pre-actions and columns representing post-actions. Each element C of the matrix describes how often the pre-action occurred before the post-action. Fig. 5a) depicts a small section of C based on a subset of 9 actions. As can be seen in the first row of the matrix, the action *Open Right Door* was frequently performed before the objects inside the right cupboard were placed on the table, e.g. in 80% of the activity recordings in which plate 1 was put on the table, the right cupboard door was opened sometime before. As mentioned before, the reason why the values or not 100% for instrumentally dependent action pairs lies in the imperfection of action recognition. The recorded activities had an exploratory nature and were not perfect demonstrations, e.g. multiple subjects opened the window even though it does not contribute to the task of setting the table. Most of them tested the opening and closing of the window only towards the end of the session. This explains why multiple values in the column under the post-action *Open Window* are 100% temporal co-occurrence, meaning that the respective actions have always been performed before the window was opened. Thus, high values of temporal co-occurrence are indicative of but not sufficient for a causal dependency between two actions. After thresholding, only a fraction of the action pairs remains (see elements highlighted in Fig. 5(a) with blue for a threshold of 0.7) and are considered as candidates for causal relationships to be tested through mental simulations. The results of testing the generated causal hypotheses of instrumental action dependencies in simulation via Algorithm 2 are shown in Fig. 5(b). The algorithm successfully finds all instrumental dependencies, e.g. that opening the right door is instrumental for putting plate 1 on the table. Besides direct, 1st-level dependencies, it can discover multi-level dependencies such as between opening the right cupboard door and putting plate 3 on the table. Since plate 3 is on the bottom of the stack of plates, it is required to first open the cupboard door, then put plate 1 and then plate 2 on the table before the action *Put Plate 3* is executable. Interestingly, our algorithm can even find 1st-level dependencies between pre- and post-actions with co-occurrence values below the chosen threshold of $\tau = 0.7$. Namely the value $C(i, j) = 0.63$ between the pre-action a_i of opening the right cupboard door and the post-action a_j of taking the third small plate. This is because the system detects that *Put Small Plate 3* is dependent on both putting plate 1 and plate 2 on the table, which are themselves dependent on opening the right cupboard door. Thus, strong short-term dependencies are used to reason about long-term dependency chains. Furthermore, the mental simulation detects that the pre-action of opening the right cupboard door makes it impossible to also open the right window, while without the pre-action, opening the window is possible. Thus, opening the right door is marked as a blocking action for opening the window in the action dependency matrix (X with orange highlighting). The elements of $\mathcal{D}$ which represent direct (1st-level) dependencies constitute an adjacency matrix from which an action dependency graph can be generated. The graph generated from the section of $\mathcal{D}$ shown in Fig. 5 b) is illustrated in Fig. 6 on the top. The numbers on the arrows represent the level of instrumental dependency from the first activity in the graph.

Algorithm 2 Instrumental Testing in Mental Simulation

Input: C: temporal co-occurence matrix
$\mathcal{A}$: list of performed actions
E: Simulation Environment
τ: Threshold
d_{max}: Maximum Dependency Level
n: number of actions
$\mathcal{D} \leftarrow$ initialize $n \times n$ dependency matrix with -1
for elements $C(i, j) < \tau$ **do**
  $\mathcal{D}(i, j) \leftarrow 0$
end for
$d \leftarrow 1$ # dependency level
while $\mathcal{D}$ contains $-1 \wedge d < d_{max}$ **do**
  for elements $\mathcal{D}(i, j) = -1$ **do**
    $s_i \leftarrow$FETCH-SEQ($a_i, \mathcal{D}$): fetch seq. of known instrumental pre-actions for a_i from $\mathcal{D}$
    $m \leftarrow$ *IsExecutable*(a_j)
    Execute s_i
    $n \leftarrow$ *IsExecutable*(a_j)
    if ¬m ∧ n **then**
      for $k \leftarrow 0$; $k < d$; $k{+}{+}$ **do**
        $l \leftarrow INDEX(s_i[k], \mathcal{A})$: get pre-action indices
        $\mathcal{D}(l, j) \leftarrow d - k$
      end for
    end if
    if m ∧ n **then**
      $\mathcal{D}(i, j) \leftarrow 0$
    end if
    if m ∧ ¬n **then**
      $\mathcal{D}(i, j) \leftarrow$X
    end if
    reset E
  end for
  $d{+}{+}$
end while
for elements $\mathcal{D}(i, j) = -1$ **do**
  $\mathcal{D}(i, j) \leftarrow 0$: ignore unresolved entries above d_{max}
end for

	Open Right Door	Put Plate 1	Put Plate 2	Put Plate 3	Put Small Plate 1	Put Small Plate 2	Put Small Plate 3	Put Wine Glass 1	Open Window
Open Right Door	0	0.80	0.79	0.82	0.84	0.80	0.63	0.73	1.0
Put Plate 1	0.20	0	0.97	0.91	0.78	0.84	0.75	0.87	1.0
Put Plate 2	0.16	0.01	0	1.0	0.57	0.82	0.63	0.87	0.5
Put Plate 3	0.02	0	0	0	0.34	0.08	0.38	0.27	0
Put Small Plate 1	0.11	0.14	0.22	0.73	0	0.94	0.75	0.53	1.0
Put Small Plate 2	0.12	0.08	0.13	0.54	0	0	1.0	0.47	0.5
Put Small Plate 3	0.04	0.03	0.05	0.3	0	0	0	0.7	0
Put Wine Glass 1	0.05	0.03	0.03	0.27	0.09	0.12	0.5	0	0.5
Open Window	0	0	0	0	0	0	0	0	0

(a) Temporal co-occurrence matrix C.

	Open Right Door	Put Plate 1	Put Plate 2	Put Plate 3	Put Small Plate 1	Put Small Plate 2	Put Small Plate 3	Put Wine Glass 1	Open Window
Open Right Door	0	1	2	3	1	2	3	1	X
Put Plate 1	0	0	1	2	0	0	0	0	0
Put Plate 2	0	0	0	1	0	0	0	0	0
Put Plate 3	0	0	0	0	0	0	0	0	0
Put Small Plate 1	0	0	0	0	0	1	2	0	0
Put Small Plate 2	0	0	0	0	0	0	1	0	0
Put Small Plate 3	0	0	0	0	0	0	0	0	0
Put Wine Glass 1	0	0	0	0	0	0	0	0	0
Open Window	0	0	0	0	0	0	0	0	0

(b) Instrumental dependency matrix $\mathcal{D}$.

Fig. 5: (a) Section of the temporal co-occurrence matrix with prior actions as rows and posterior actions as columns. The matrix elements represent the ratio of human activities in which our action recognition system inferred that the prior action was executed before the posterior action. The highlighted elements have co-occurrence values higher than the threshold of 70% and are thus the causal candidates that are tested for instrumental dependency in mental simulation.
(b) Section of the instrumental dependency matrix where 1 represents a 1st-level instrumental dependency between pre- and post-action, 2 and 3 represent 2nd-level and 3rd-level dependencies, 0 represents no dependency and X marks that the prior action renders the posterior action non-executable.

For evaluation, we compare the performance of our system using human activity data to find candidates for mental simulation against the brute-force approach where no prior information is used and therefore all possible pairs of actions in the environment need to be tested. The brute-force approach equates to the case where the temporal co-occurrence threshold is set to 0. Fig. 7 depicts that in this case all instrumental dependencies are found with a runtime of $1023s$ [3]. Using a higher threshold strongly decreases the runtime, since fewer candidates need to be tested in simulation, while no drop in the number of discovered dependencies is observed until $\tau = 0.8$, where the percentage of discovered dependencies drops to 77% and 47% for both $\tau = 0.9$ and $\tau = 1.0$. The latter represents the case in which the system only tests pre-post action pairs which co-occurred in the right temporal sequence in 100% of trials. Because there are misclassifications in the action recognition system, several ground-truth dependencies have temporal co-occurrence values of less than 1.0, are thus not considered as potential candidates for simulation, and therefore not

[3]The analysis has been done on a system using an Intel Core i78565U CPU, 16 GB RAM, and an Nvidia GTX1080 GPU.

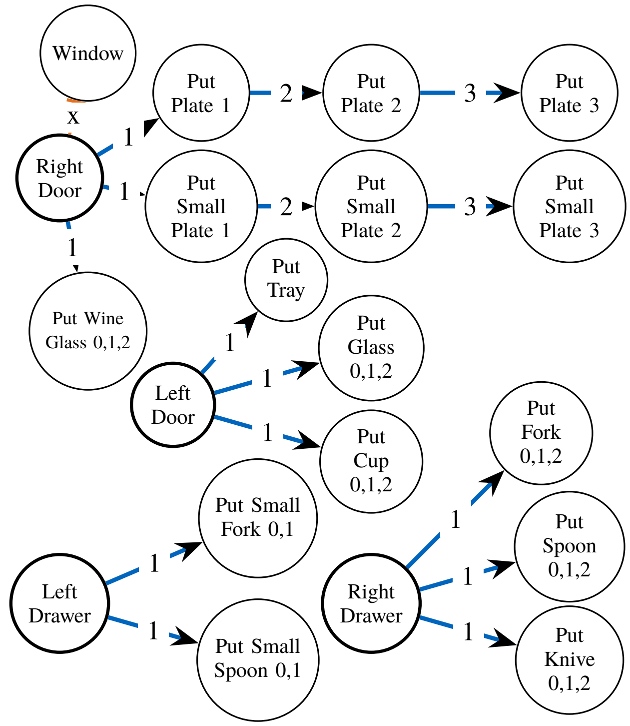


Fig. 6: An overview of all 30 action dependencies present in the *Setting a Table* scenario. Most actions depend only on one other instrumental action, while some actions require the prior execution of two (plate 1, small plate 1) or three (plate 2, small plate 2) other actions. The method also detects that opening the right door prevents opening the window, here depicted with an orange X arrow.

discovered. For our scenario and perception system, we consider the threshold of $\tau = 0.7$ as optimal due to allowing the system to discover all instrumental action dependencies while reducing the runtime to $168s$ - a 6-fold reduction compared to the brute-force approach.

V. DISCUSSION

Our results support the hypothesis that following the scientific approach of observation, generation of causal hypotheses, and deliberate testing enables a cognitive system to efficiently construct knowledge about acting effectively in its environment. Even with a limited set of 39 possible actions in the *Setting a Table* environment, our method was six times faster in finding all action dependencies than the brute-force approach of testing all possible pairs. Since the brute-force approach scales proportional to the number of permutations of action sequences in the environment, we expect an even higher comparative gain in computational speed for more complex environments. Higher accuracy in action recognition would allow to set a higher threshold τ and therefore sorting out a higher percentage of action pairs before mental simulation, thus further increasing the run-time reduction of our method.

We have tested our approach in a single environment and task. To generalize this causal inference to other environments and tasks, abstract representations will be necessary. One possible approach is using ontologies and considering environmental states as prerequisites for certain actions to learn planning models [20]. Environment-specific action de-

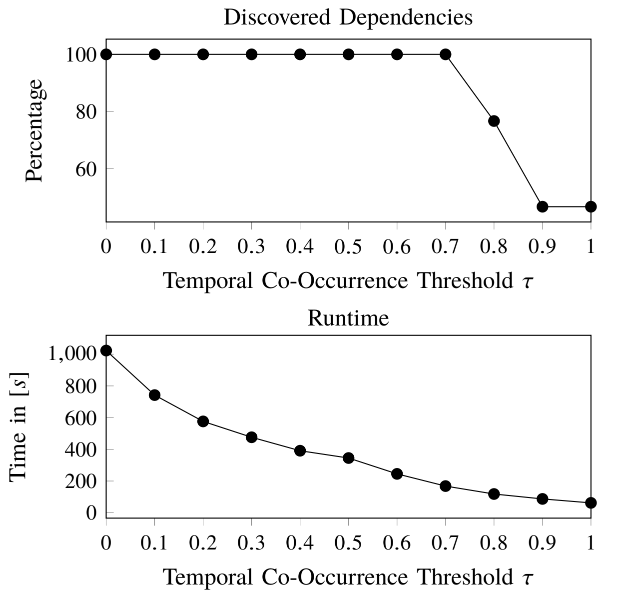


Fig. 7: The temporal co-occurrence threshold decides which pre-post action pairs are considered as candidates for action dependencies and are tested in mental simulation. The boundary case $\tau = 0$ represents the brute force approach in which all action pairs are considered as candidates and is shown to find all dependencies at the cost of a running time of $1023s$. The boundary case $\tau = 1$ represents the case in which only action pairs where the pre-action occurred before the post-action in 100% of trials are considered as candidates. Because of the uncertainty of action recognition, in this case, not all action dependencies are discovered, but the run-time decreases to $62s$. In our scenario, the optimal run-time/performance ratio can be found at a threshold of $\tau = 0.7$. All action dependencies are discovered with a run-time of $168s$, corresponding to a 6-fold increase in speed compared to the brute force approach.

pendency graphs such as those constructed by our method could be used to check automatically generated action sequences for executability and correct them by adding missing instrumental actions.

VI. CONCLUSION

In this work, we have shown that a dataset of multiple human goal-oriented activities can be used for bootstrapping exploratory testing of activity dependencies in an environment. We show that our method can significantly decrease the required computation time without loss in performance. This approach of creating cognitive systems that combine observations of human activities with the experimental testing of causal hypotheses resembles the scientific approach in which scientists and arguably also children make sense of the world. This is achieved by observing and subsequent testing in which actions are instrumental for the execution of other actions. Future research will focus on two main themes - generalization and embodiment. We plan to apply our method to other environments and investigate how knowledge generated in one environment can be used to bootstrap the learning of action dependencies in another environment. In humans, the verification of whether a certain action is executable depends on the individual's embodiment and the related perception of affordances. Therefore, these constraints should be taken into consideration when testing hypotheses in mental simulation or in a physical environment.

ACKNOWLEDGMENT

The research reported in this paper has been partially supported by the German Research Foundation DFG, as part of Collaborative Research Center (Sonderforschungsbereich) 1320 "EASE - Everyday Activity Science and Engineering", University of Bremen (http://www.ease-crc.org/). The research was conducted in subproject R1: NEEM-based embodied knowledge system.

References

[1] C. Cook, N. D. Goodman, and L. E. Schulz, "Where science starts: Spontaneous experiments in preschoolers' exploratory play," *Cognition*, vol. 120, no. 3, pp. 341–349, 2011.
[2] G. Cheng, K. Ramirez-Amaro, M. Beetz, and Y. Kuniyoshi, "Purposive learning: Robot reasoning about the meanings of human activities," *Science Robotics*, vol. 4, no. 26, p. eaav1530, 2019.
[3] J. Haugeland, *Artificial intelligence: The very idea*. MIT press, 1989.
[4] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent advances in robot learning from demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, 2020.
[5] S. Schaal, "Learning by demonstration," in *Encyclopedia of Machine Learning and Data Mining*, 1996.
[6] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics Auton. Syst.*, vol. 54, pp. 361–369, 2006.
[7] A. Haidu and M. Beetz, "Action recognition and interpretation from virtual demonstrations," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2833–2838, 2016.
[8] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object–action relations by observation," *The International Journal of Robotics Research*, vol. 30, pp. 1229 – 1249, 2011.
[9] D. Vernon, C. von Hofsten, and L. Fadiga, "Desiderata for developmental cognitive architectures," *Biologically Inspired Cognitive Architectures*, vol. 18, pp. 116–127, 2016.
[10] D. C. Bentivegna, C. G. Atkeson, A. Ude, and G. Cheng, "Learning to act from observation and practice," *International Journal of Humanoid Robotics*, vol. 1, no. 04, pp. 585–611, 2004.
[11] T. Inamura, T. Shibata, H. Sena, T. Hashimoto, N. Kawai, T. Miyashita, ..., and Y. Yoshikawa, "Simulator platform that enables social interaction simulation — sigverse: Sociointelligenesis simulator," in *2010 IEEE/SICE International Symposium on System Integration*.
[12] T. Bates, K. Ramirez-Amaro, T. Inamura, and G. Cheng, "On-line simultaneous learning and recognition of everyday activities from virtual reality performances," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3510–3515.
[13] 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.
[14] R. King, K. Whelan, F. Jones, P. Reiser, C. Bryant, S. Muggleton, D. Kell, and S. Oliver, "Functional genomic hypothesis generation and experimentation by a robot scientist," *Nature*, vol. 427, no. 6971, pp. 247–252, 2004.
[15] J. Pearl, M. Glymour, and N. Jewell, *Causal inference in statistics: An overview*. Wiley, 2009.
[16] P. Tadepalli, C. Barrie, and S. J. Russell, "Learning causal trees with latent variables via controlled experimentation," 2019.
[17] M. Tenorth, J. Bandouch, and M. Beetz, "The tum kitchen data set of everyday manipulation activities for motion tracking and action recognition," *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pp. 1089–1096, 2009.
[18] K. Ramirez-Amaro, C. Uhde, T. Bates, and G. Cheng, "A benchmarking dataset for automatic symbolic grounding from virtual demonstrations," in *2nd International Workshop on Computational Models of Affordance in Robotics (ICRA)*, 2019.
[19] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[20] M. Wächter, S. Schulz, T. Asfour, E. Aksoy, F. Wörgötter, and R. Dillmann, "Action sequence reproduction based on automatic segmentation and object-action complexes," in *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2013, pp. 189–195.