A Causal Approach to Tool Affordance Learning

Jake Brawer[1,*] and Meiying Qin[1] and Brian Scassellati[1]

***Abstract*— While abstract knowledge like cause-and-effect relations enables robots to problem-solve in new environments, acquiring such knowledge remains out of reach for many traditional machine learning techniques. In this work, we introduce a method for a robot to learn an explicit model of cause-and-effect by constructing a structural causal model through a mix of observation and self-supervised experimentation, allowing a robot to reason from causes to effects and from effects to causes. We demonstrate our method on tool affordance learning tasks, where a humanoid robot must leverage its prior learning to utilize novel tools effectively. Our results suggest that after minimal training examples, our system can preferentially choose new tools based on the context, and can use these tools for goal-directed object manipulation.**

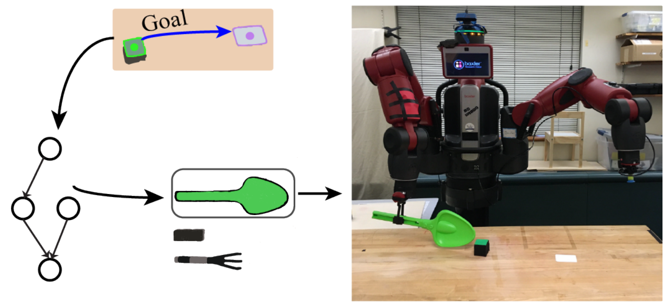


Fig. 1: Tool selection via causal inference. A Baxter collaborative robot queries a learned causal model of tool-assisted manipulation using perceptual information from its workspace. Information from the graph is used to select the most appropriate tool for completing its goal.

I. Introduction

In *The Crow and the Pitcher*, the fifth century B.C.E. Greek poet Aesop wrote of a thirsty crow who ingeniously found relief after dropping stones into a jug of water until the water level was high enough for the crow to drink from it. More recently, the scientific community corroborated this remarkable feat of physics-based instrumental problem-solving in New Caledonian Crows (NCC; *Corvus moneduloides*) [1]. Contrary to the flexible reasoning capabilities of crows and other higher species, robot learning and reasoning remain starkly limited despite substantive advancements in statistical machine learning techniques. Additionally, what exactly is learned by these systems remains largely opaque [2], which not only makes them difficult to debug but poses serious risks in robotics settings where unforeseen behaviors can cause physical harm.

Ideally, robots should be able to acquire knowledge and skills in a way that is both transparent and portable across contexts, but without compromising on the incredible advancements made by the machine learning community. To that end, we have developed a system whereby a robot can learn and exploit a graphical causal model to solve a physical reasoning task. Our approach has a robot learning a structural causal model (SCM) [3] through a mix of observation and physical exploration. The SCM is used to perform causal inference, which is completed by a group of neural networks that are dynamically constructed and trained as a function of the learned structure of the SCM and the goals of the current task. As a result, our system represents the robot's knowledge in an explicit and explainable way by the directed acyclic graph (DAG) entailed by the SCM, but that also leverages the powerful pattern recognition capabilities of machine learning techniques via the use of neural networks.

We demonstrate our method on a humanoid robot that must build a model of the cause-and-effect relationships underlying tool-assisted manipulation and then use this model to both solve goal-directed manipulation tasks, and quickly learn the affordances of novel tools given a kinematic model of each tool. We demonstrate that our system is capable of selecting the appropriate tool and action to take to move an arbitrarily placed block into goal regions, as well as leveraging prior learning to bootstrap learning of new tools.

In sum, our contributions are the following:

- A method for a robot to construct a transparent model of causation via a mix of observation and experimental learning.
- A method for performing forward and backward causal inference using a series of dynamically constructed neural networks.
- A method for a humanoid robot using learned causal models to quickly learn and utilize tool affordances of novel tools.

II. Related Work

A. Causality in ML and Robotics

Many researchers have attempted to formalize causal relations over the past century. Here we focus on Pearl's SCM framework to formalize causal relations using directed acyclic graphs (DAGs) that define structural equations between causal variables [3]. Pearl argues that SCMs accommodate sophisticated forms of reasoning, including interventional (e.g., "What if I had done X?") and counterfactual (e.g., "What if I had acted differently?") [4]. Counterfactual

*Jake Brawer is the corresponding author (`jake.brawer@yale.edu`).

[1]Department of Computer Science, Yale University, 51 Prospect St., New Haven, CT, USA.

reasoning has been integrated into RL systems; for example, it has been shown to enable the acquisition of effective decentralized multi-agent policies in a credit assignment task [5], and aid in the evaluation of high-risk healthcare policies [6].

SCM-based reasoning has been employed in a robot by Angelov et al. [7] which used learned SCMs to counterfactually reason about user preferences in their demonstrations of a motion task. Non-SCM based approached to causal reasoning include Xiong et al. [8], which integrated spatial, temporal, and causal and-or graphs learned from user demonstrations to enable a robot to fold clothing. Nevertheless, causal learning and reasoning for robots remains relatively unexplored despite the active developments from the machine learning and artificial intelligence communities.

B. Affordance learning

Ecological psychologist J.J. Gibson defined object affordances as the possible actions an agent can take on object as a function of both the properties of the object and the physiology of the agent [9]. For example, a coffee mug is "pick up-able" to humans, and not to dolphins, because its shape fits nicely in the human hand. While there have been many approaches to formalizing affordances in a robotics context (see [10] for a review), a common framework defines affordances as the three-way relationship between an object, the actions a robot can take, and the effects of these actions on the object [11]. While Bayesian networks have been a popular method for representing affordances (see [12] for a recent example), an explicitly causal framework is rarely employed despite centrality of cause and effect within many affordance paradigms. As frameworks like SCMs can tease apart causal, and not merely correlational, properties in the environment, a robot equipped with such a framework is better poised to model the effects of its actions on the world, and thus to acquire and utilize affordances. We demonstrate this capability by showing that our system can effectively complete goal-directed tool and action-selection tasks using acquired affordance knowledge.

III. Problem Statement

Our goal is to have a robot model simple tool use behaviors using SCMs, and then subsequently use this model to quickly ascertain the affordances of novel tools. We begin by giving a brief overview of the SCM formalism (refer to [3] for more technical details).

The problem of learning an SCM is twofold: the structure of the graph which links variables to other variables – or causes to effects– must be learned, as well the functional mechanism that formalize the these relationships.Formally an SCM $\mathcal{M}$ is a tuple $\{\mathcal{U}, \mathcal{V}, \mathcal{F}\}$ where $\mathcal{U}$ is a set of unobserved background features or, "exogenous" variables, $\mathcal{V}$ is a set of observed features, or "endogenous" variables, and $\mathcal{F}$ is a set of functions that assigns values to variables in $\mathcal{V}$ based on other variables in $\mathcal{U}$ and $\mathcal{V}$. Under the assumption that the causal structure is acyclic, there is a corresponding DAG $\mathcal{G}$ where the nodes in the graph correspond to variables in $\mathcal{U}$ and $\mathcal{V}$ and the edges to the functions in $\mathcal{F}$.

Kocaoglu et al. [13] demonstrated that feedforward neural networks admit an interpretation as SCMs. We take advantage of this fact by using the causal structure uncovered during learning to guide the construction of series of neural networks that define the structural equations $\mathcal{F}$ underlying the SCM. The advantage of using such a scheme it minimizes the assumptions of the generating distributions of the variables of interest, which in many potential real-world scenarios are likely unknown a priori. Exogenous variables $\mathcal{U}$ are represented by the hidden nodes in the neural network.

IV. Our approach

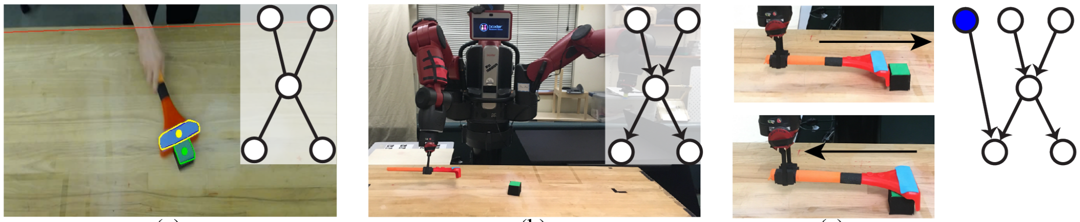
A. Causal Learning

The goal of the **observational phase** is to bootstrap learning of causal structure by taking advantage of the fact that it can be inferred (usually within an equivalence class of DAGs under reasonable assumptions of causation, see [14] for an overview of causal discovery) from passively collected, observational data using standard structural learning algorithms. This helps minimize the amount of interventional data required to fully specify the graph, which may be beneficial, as often collecting this sort of data is difficult to collect for reasons of practicality. The result of this phase is an undirected graph with edges drawn between causally dependent nodes, though the direction of causation may not be known.

Subsequently, the robot enters the self-supervised **validation phase**, wherein it attempts to orient the graph by collecting interventional data. An edge between nodes V_1 and V_2 is oriented $V_1 \rightarrow V_2$ if $P(V_2|do(V_1 = x) \neq P(V_2|do(V_1 = y)$ for some interventions x and y, and vice-versa. As interventional samples can be difficult or costly to obtain, it is desirable to be able to preferentially select nodes to intervene on such that the total number of experiments is minimized. To that end we take an active learning approach adapted from [15], which simply intervenes on nodes in the graph produced during the observational phase starting with the nodes with the highest degree, ensuring that the most informative interventions are carried out first.

Finally, in the **augmentation phase**, the robot attempts to incorporate a new node V_3 into its causal model. Unlike in the validation phase, the goal of the augmentation phase is not orienting edges that already exist, but rather to add new oriented edges where none had existed previously. An edge between a new node V_3 and an extant node V_1 is added to the model if it is determined that intervening on V_3 effects the value of V_1 using the method described above.

In order to minimize the number of edges to be tested, and thus the chances of falsely identifying causal relationships, we developed a few heuristics for selectively testing nodes. First, nodes are tested in topological sorted order. This ensures that the causal antecedents that have thus far been identified are tested first. In addition, we make the following assumption: if an edge $V_1 \rightarrow V_2$ is found for some nodes V_1, V_2, then we do not test any descendants of V_2. While



(a) (b) (c)

Fig. 2: The causal discovery Process. During the observation phase (a) the robot learns a skeleton of the causal graph observing demonstrations performed by a human. During the validation phase (b) the robot attempts to orient the edges of the graph via self-supervised experimentation. Finally, during the augmentation phase (c), the robot introduces a new node (blue) and attempts to incorporate it into its graph via further experimentation.

this has the potential to produce an incomplete causal graph, it avoids the possibility of mistaking indirect causation from $V_1 \rightarrow V_2 \rightarrow V_3$ for some descendant V_3 of V_2, for the direct causal connection $V_1 \rightarrow V_3$.

B. Causal Inference

Once the causal structure is in place, the functional mechanisms underlying the SCM can be learned, allowing for the robot to perform causal inference. As we wish to minimize our assumptions about these underlying mechanisms, we use neural networks to estimate these functions, using the structure of the graph to guide the structure of the neural network architectures. However, while this allows for reasoning from causes to effects, it is not immediately clear how other forms of causal inference, e.g. diagnostic or 'abductive' reasoning from effects to causes, can be performed. Similarly, in many real-world scenarios, it is often the case that observations have been made for only a subset of variables in the SCM, and it may be desirable to estimate the unknown values using known information.

In order to support these capabilities, the final set of neural network architectures are dependent on not only the causal graph structure G, but the observed and unobserved variables as well. We treat these unobserved values as a set of "queries" Q, the values of which we would like to estimate. Each node $q \in Q$ is ranked with "causal score" that is used to ultimately guide the order of inference, as well the construction of the neural networks via a recursive algorithm (refer to algorithm 1). A causal score is a value between 0 and 1, computed as the ratio of the number of q adjacent nodes with observed values to the total number of q adjacent nodes. If the values of all of q's parents are observed, it is automatically assigned a score of 1, as it's value can be estimated by standard means. Once the scores are computed, the value of the node with the highest score is inferred, and it is treated as an observed variable, and so can be used to infer the values of other qs. This process of scoring and inference is repeated until all qs have been estimated.

Inferring the q values was accomplished using feedforward neural networks, though the structure of these networks may not match the forward structure of the corresponding causal graph in cases where abductive inference is required. We use two heuristics to construct each network f; 1) if all of q's parents are observed, then $q = f(Pa(q))$, where $Pa(X)$ denotes the parent nodes of node X; 2) otherwise, $q = f(V_{observed} \cup V_{collider})$, where $V_{observed}$ is the subset of nodes adjacent to q that have been observed, and $V_{collider}$ is the possibly empty subset of nodes that belong to a collider subgraph with q, as these dependencies may carry useful information about q's state. Thus the size of the input and output layers of any neural network was equal to the number of nodes in $V_{observed} \cup V_{collider}$ and the number of nodes in q (i.e. 1), respectively. In practice, we found that a single hidden layer consisting of 10 nodes worked well across all networks.

Algorithm 1 Recursive procedure for causal inference.

procedure ANSWERQUERIES(Q, G)
  if Q is empty **then return**
  else
    $(q, score) = max(causalScores(G, Q))$
    if $score == 1.0$ **then**
      $Input = Pa(q)$
    else
      $Input = V_{observed} \cup V_{colliders}$
    $\hat{q} = f(Input)$
    Append $\hat{q}$ to $G.obs$
    $AnswerQueries(Q \setminus q, G)$

C. Affordance learning

Andries et al. [12] identify equivalences between affordances based on how similar their effects are on an object when acted upon. We employ a similar approach to learn and represent tool affordances. For some tool t and set of possible actions A, we represent its affordances as a vector $\mathbf{a^t} \in R^{|A|}$, where each element a_i^t of the vector corresponds to a measure of how well the effects of a tool action align with predictions (here we use the coefficient of determination, r^2). Using a learned causal model of an exemplary tool, a robot can quickly estimate these vectors from a small number of exploratory actions on an object and use these estimates for tool selection and usage given some goal position for the target object.

V. Materials and Methods

In this section we discuss the design and implementation details of the task used to evaluate our model.

A. The Task

We would like the robot to 1) model the causal process underlying manipulating objects with tools, and 2) leverage the learned model to more effectively learn and wield new tools. Here we take as examples pushing and pulling as classes of actions the robot can perform. We assume the robot can observe the random variables corresponding to the initial position and final position of a manipulated block, p_{init} and p_{final}, respectively, as well as the movement vector d, which is defined as the vector that the tool tip travels starting from a small displacement before or after the block to the terminus of the push action. While $p_{init}, p_{final}, d \in R^2$, we represent each of their dimensions by their own individual nodes, which we express with a superscript, for example, p^x_{init} and p^y_{init} For convenience, when we omit the superscript, we are referring to both dimensions simultaneously. Additionally, while $d \in R^2$, for simplicity, we limit the push vectors the robot can enact to 12 evenly spaced vectors around the unit circle. This requires that for any estimated value of d, the robot must choose one of these 12 vectors with the smallest angular distance to d to enact. The robot can take actions $push, pull \in A$ in directions d, though does not at the outset know the relationship between these two, or any other variables. Consequently, during inference, d and A are estimated using classifier networks,while p_{init} and p_{final} are estimated using regressor networks.

While it is assumed the robot knows how to perform simple pushes and pulls with each tool, it does not know the effects that tool use entails, nor in what scenarios a given tool is appropriate. While the robot only needs to learn the causal relationships among these 7 variables, the number of possible DAGs is super-exponential in the number of variables [16], for a total of approximately 1.1×10^9 possible DAG structures, making this structural identification problem non-trivial.

In addition there are a number of details related to the implementation of three learning phases that are specific to this particular task, outlined below.

Observational phase: The robot observed demonstrations of a block being pushed with a hoe tool (see Figure 3) by a human, and tracks p_{init}, p_{final}, and d. This observational data is standardized and passed to a structural learning algorithm in order to get an initial hypothesis of the causal structure. For the purposes of this experiment we use the PC algorithm, a widely used score-based method for for causal discovery [17].

Validation phase: We allow the robot to directly intervene on p_{intial} p_{final} and d, and limit it only to pushes with the hoe from the previous phase. Each intervention is treated as a randomized controlled trial, where the intervened variable is forced to take on one of two values. Interventions on p_{init} and p_{final} compare interventional distributions for two prespecified positions, where as interventions on d compares the distribution induced by taking a random action vs taking no action at all. As we wish to limit the assumptions we make about the underlying distributions of the variables, we use the Kolmogrov-Smirnoff test to nonparametrically estimate difference between the two interventional distributions. Given that these interventions are used to quickly infer causal relations, and are not themselves rigorous scientific experiments, the $p < .05$ significance convention need not be followed. Here we relax the threshold of significance to $p < .2$, though this may be treated as a hyperparamter trades-off risk of type I errors for data-efficient estimation.

Augmentation phase: During the augmentation phase, the robot attempts to add the action type A to its causal model. This proceeds in much the same way as the validation phase, except we allow the robot to perform pulls as well as pushes.

B. Affordance-based tool selection

We would like the robot to choose the best tool, $\hat{t}$, and tool usage $\hat{d}$ given the circumstances of the environment. Using desired movement vector d^*, obtained by querying the causal graph, together with the estimated affordance information $a_x^{t_i}$ for a given tool t_i, the robot can make this selection. This is captured by the heuristic

$$\hat{t}, \hat{d} = \underset{t_i, d_{t_i}}{\arg\min}(2 - a_x^{t_i})(2 - \cos\theta_{d_{t_i} d^*}) \quad (1)$$

Here d_{t_i} is a movement direction a robot is capable of actuating with tool t_i according to its kinematic model. In essence, this heuristic chooses a tool based on the trade-off between how well it affords a desired action in general, and how well it can enact the action in this specific instance.

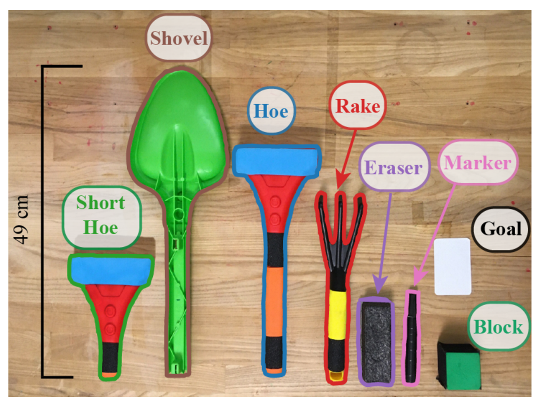


Fig. 3: The tool set.

C. The workspace

Experiments were performed on a Baxter collaborative robot (Fig 1)[1]. Our Baxter model was equipped with a claw gripper for manipulating the tools, as well as a suction gripper for picking and placing the block during the self-supervised learning phases. Informal tests suggested our model's end-effector precision was within $\pm 1cm$. Figure 3 depicts the contents of the robot's work space. Actions were

[1]Source code can be found at https://github.com/ScazLab/crow

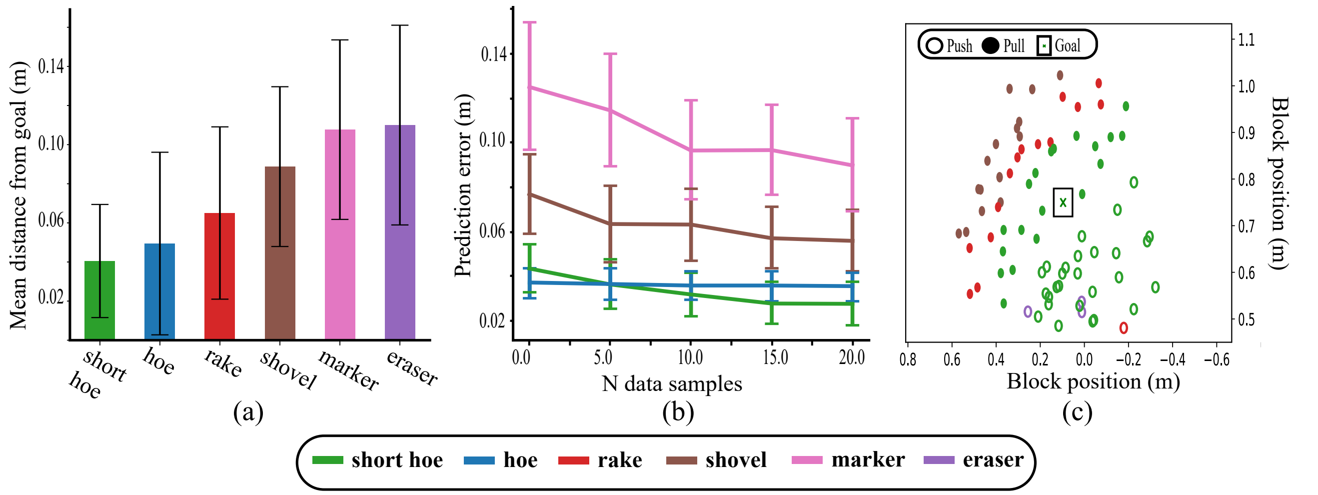


Fig. 4: Tool reasoning results. a) depicts the mean distance of the center of the block to the center of the goal region for each tool. b) depicts the learning curve for a select number of tools given initial training on the hoe. c) depicts how tools were selected and used as a function of the block's position given a static goal.

performed on a $5cm^3$ wooden block. Initially the robot had access only to a plastic hoe measuring $12.5cm$ from grip point to tool tip. Additionally, there were 5 morphologically distinct tools the robot's learning was evaluated on. During the evaluation phase, the robot was tasked with pushing the block into a small rectangular goal region measuring approximately, $10cm \times 8.5cm$. The positions of the tool tips, the block, and goal region were tracked with a head-mounted RGB webcam using a blob detector calibrated to the colors unique to each object. We used the Causal Discovery Toolbox's [18] implementation of the PC algorithm, as well as classifier and regressor neural network implementations from Scikit-learn [19].

D. Evaluation

With our experiments we wished to see how well a robot could use a learned causal model to perform affordance-based reasoning for familiar and unfamiliar tools. In order to do this, we ran three evaluations: 1) Given 8 samples with each of the 5 novel tools, we had the robot choose both the best tool and action to perform given a single fixed goal location, but arbitrary initial positions of the block; 2) given 20 training samples per tool, multiple goal regions and multiple initial block positions, we observed how close could the robot move the block towards the goal region for each tool; 3) we looked at how accurately the robot could predict action effects as a function of training data samples for a subset of the 6 tools.

VI. Results

Figure 5 depicts the learned causal graph. The structure of this graph models two important aspects of the task: 1) the final position of the block is a function of the initial position of the block and the direction in which it is pushed, and 2) the type of action, push or pull, effects the push direction, suggesting the robot has successfully grounded these abstract actions to a concrete sensorimotor effects. The observation phase consisted of 60 sample demonstrations of a human pushing a block. During the validation and augmentation phases, we limited interventions to 20 samples per intervention. During the validation phase the robot performed two interventions, for a total of 40 samples. The augmentation phase consisting of one intervention on action type, consisted of 20 samples.

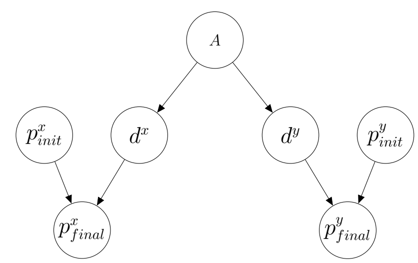


Fig. 5: The learned structure of the SCM.

Figure 4(a) depicts the robot's performance with each tool across 5 goal locations with 5 randomly distributed initial positions per goal location. Remarkably, the top 3 best performing tools, short hoe ($M = 0.04, SD = 0.02$), the hoe ($M = 0.5, SD = 0.04$), and the rake ($M = 0.06, SD = 0.04$), came within a few centimeters of the goal on average, despite relative inaccuracy of the robot and the limited set of movement directions at the robot's disposal. The tool performance roughly track with morphological similarity with the base tool (the hoe), with the most similar tools producing the best results (refer to Figure 3). The learning curve results depicted in (b) tell a similar story, as the tools that are morphologically most similar to to original tool– the short hoe, and to a lesser extent, the shovel– benefit much more from the prior learning than the marker. Interestingly, the short hoe ultimately performs better than the original tool, though this is likely due to the fact that the short hoe is physically the same tool as the original hoe, but just gripped closer to the tool tip, producing more consistent pushes and pulls.

Figure 4(c) depicts how our system chose tools and actions as a function of the object's initial position given a fixed goal location. Here we see action selection in line with expectations; the robot chose to push the block when the block was positioned before the goal relative to the robot, and pulled it when it was positioned beyond the goal, further supporting supporting the notion that the system has meaningfully grounded the push and push actions to sensorimotor effects. Figure 4(a) also helps shed light on the tools choices in (c). The robot overwhelmingly preferred the short hoe for pushing and pulling, which makes sense as it is the tool the robot uses most effectively. On the periphery of the workspace, the robot begins choosing the rake for pulling, as the rake is both slightly longer than the short hoe, and also the robot's next best tool. The same reasoning for the choice of shovel on the edges of the workspace; it is the longest tool available to the robot and thus the only one capable of completing the desired pull.

VII. Discussion and Conclusion

In this work, we demonstrated a method for a robot to learn and utilize a causal structural model to rapidly acquire and reason over tool affordances. The results of our experiments suggest that the grounding of actions to effects via the learned causal model enabled the robot to effectively select and use tools preferentially based on the conditions of the workspace. We believe there are two primary advantages of this method over more common approaches to robot learning: 1) The knowledge acquired through the learning process is explicitly represented and hierarchically organized by virtue of the DAG structure of the SCM; 2) By leveraging this same DAG structure to construct neural networks, the functional mechanisms of the SCM can be learned in relatively data-efficient way. That is, even in a dense causal network, for a given node of interest, only a subset of the nodes are required to infer the node's value, mitigating the effects of the curse of dimensionality that often plagues machine learning systems. The transparency of knowledge entailed in 1) is vital in a robotics context, for example in human–robot collaborative contexts where shared expectations amongst collaborators has been demonstrated to be important for team success [20].

Nevertheless, there were some limitations to this work. There were relatively few causal variables under consideration, and all of them were assumed to be observable. Currently, it is not clear how well our method would scale to learning larger, more complex causal graphs or graphs with latent causal variables. In addition, aspects of the interventional experiments conducted by the robot, including the two initial positions of the block, were hard-coded. Ideally, the robot should be able to autonomously be able to generate its own experiments and choose values with which to force the interventional variables to take on. This is an important problem, as depending on the generating distribution underlying the model, some interventions may be more informative than others for uncovering causal relationships.

VIII. Acknowledgments

This work was supported by the Office of Naval Research award #N00014-18-1-2776.

References

[1] S. A. Jelbert, A. H. Taylor, L. G. Cheke, N. S. Clayton, and R. D. Gray, "Using the aesop's fable paradigm to investigate causal understanding of water displacement by new caledonian crows," *PloS one*, vol. 9, no. 3, p. e92895, 2014.

[2] G. Marcus, "Deep learning: A critical appraisal," *arXiv preprint arXiv:1801.00631*, 2018.

[3] J. Pearl, *Causality: models, reasoning and inference*. Springer, 2000, vol. 29.

[4] ——, "Theoretical impediments to machine learning with seven sparks from the causal revolution," *arXiv preprint arXiv:1801.04016*, 2018.

[5] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] M. Oberst and D. Sontag, "Counterfactual off-policy evaluation with gumbel-max structural causal models," *arXiv preprint arXiv:1905.05824*, 2019.

[7] D. Angelov, Y. Hristov, and S. Ramamoorthy, "Using causal analysis to learn specifications from task demonstrations," in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 1341–1349.

[8] C. Xiong, N. Shukla, W. Xiong, and S.-C. Zhu, "Robot learning with a spatial, temporal, and causal and-or graph," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2144–2151.

[9] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Boston:Houghton Mifflin, 1979.

[10] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational models of affordance in robotics: a taxonomy and systematic classification," *Adaptive Behavior*, vol. 25, no. 5, pp. 235–271, 2017.

[11] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Modeling affordances using bayesian networks," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 4102–4107.

[12] M. Andries, R. O. Chavez-Garcia, R. Chatila, A. Giusti, and L. M. Gambardella, "Affordance equivalences in robotics: a formalism," *Frontiers in neurorobotics*, vol. 12, p. 26, 2018.

[13] M. Kocaoglu, C. Snyder, A. G. Dimakis, and S. Vishwanath, "Causalgan: Learning causal implicit generative models with adversarial training," *arXiv preprint arXiv:1709.02023*, 2017.

[14] F. Eberhardt, "Introduction to the foundations of causal discovery," *International Journal of Data Science and Analytics*, vol. 3, no. 2, pp. 81–91, 2017.

[15] A. Hauser and P. Bühlmann, "Two optimal strategies for active learning of causal models from interventional data," *International Journal of Approximate Reasoning*, vol. 55, no. 4, pp. 926–939, 2014.

[16] R. Robinson, "Counting labeled acyclic digraphs." in *New Directions in the Theory of Graphs: Proc of the Third Ann Arbor Conf. on Graph Theory*. Academic Press, 1973, pp. 239–273.

[17] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.

[18] D. Kalainathan and O. Goudet, "Causal discovery toolbox: Uncover causal relationships in python," *arXiv preprint arXiv:1903.02278*, 2019.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[20] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," *learning*, vol. 8, no. 9, 2013.