Learning Low-Level Causal Relations using a Simulated Robotic Arm

Anonymous submission

No Institute Given

Abstract. Causal learning allows humans to predict the effect of their actions on the known environment and use this knowledge to plan the execution of more complex actions. Such knowledge also captures the behaviour of the environment and can be used for its analysis and the reasoning behind the behaviour. This type of knowledge is also crucial in the design of intelligent robotic systems with common sense. In this paper, we study causal relations by learning the forward and inverse models based on data generated in a simulated robotic arm involved in two sensorimotor tasks. As a next step, we investigate feature attribution methods for the analysis of the forward model, which reveals the low-level causal effects corresponding to individual features of the state vector related to both the arm joints and the environment features. This type of analysis provides solid ground for dimensionality reduction of the state representations, as well as for the aggregation of knowledge towards the explainability of causal effects at higher levels.

Keywords: causality · forward model · inverse model · feature importance · explainability

1 Introduction

Observing and learning causal relations in a given environment is an essential element of cognition in humans and other high animals. Thanks to this ability, agents can form intuitive theories from multiple observations and use them to predict the environment's behaviour in response to their actions [4]. This common sense understanding includes the knowledge of intuitive physics, a key ingredient of early cognitive development [9].

Causal models capturing and learning causal relationships from observations can be used for action planning towards task completion. Analysis of these models encapsulating intuition about the given environment and their predictions can be helpful for causal reasoning. In humans, a seven-grade model of the evolution of causal cognition has been proposed, with increasingly more complex causal skills [11]. These range from individual causal understanding and tracking behaviour (understanding perceived effects of one's own motor actions) up to what the authors call causal network understanding (e.g. after learning that wind can cause an apple to fall, the person may understand that wind can also cause other things to fall or move) [6].

Studying causation is a complex area, and it is being transferred to robotics, which essentially involves embodied agents (robots) interacting with the world [7]. The authors proposed an analysis of the role of causal reasoning in robotics, organized into two parts. The first part is a novel categorization of robot causal cognition inspired by the categorization of human causal cognition [11]. The latter describes a hierarchy of seven grades of causal skills, with humans mastering all grades and animals only certain grades, all according to their stage in evolution. They define eight categories of robot causal cognition at the sense–plan–act level, divided into three groups: learning causal relations, inferring the causes related to an interacting human, and robot deciding how to act. In our work, we focus on two categories of learning causal relations, as explained below.

Studying causality is an emerging area of machine learning research [22, 18], fueled by inspiration from the pioneering work on graphical causal inference by Judea Pearl [15] and collaborators [16]. It has been argued that understanding causality can be beneficial for machine learning and artificial intelligence, leveraging it toward building more robust models with common sense [18, 23]. In machine learning, the conceptual framework deals with formal systems that can be studied at four levels as proposed in [18], ranging from the most detailed mechanistic/physical level, through structural causal, causal graphical, up to (most abstract) statistical level.

The ideas to causal mechanisms are also naturally usable in robotics, where these levels apply. In our work, we focus on the lowest, mechanistic level that corresponds to individual components of the state space representation, including the robot's individual joints and features of the world. Our work offers two main contributions: (1) we explore causal learning using forward and inverse models encapsulating intuitive knowledge about the environment and train them on synthetic data generated by motor babbling in simulation, and (2) we analyze these models to extract the information about the behaviour of the environment.

2 Background and Related Work

The sensorimotor knowledge in a robotic system is commonly represented by a pair (or pairs of) of complementary models: the *forward model* (FM) that predicts sensory consequences of one's own actions, and an *inverse model* (IM) that predicts actions in order to reach the desired state [21].[1]

The FM is commonly called a causal model, which is mathematically well-defined, whereas the IM is non-causal since it solves an inverse problem where the causes (actions) and effects (states) are temporally reversed. Mathematically, inverse kinematics is an ill-posed problem in general since in redundant robots many actions (solutions) can lead to the desired state. As explained in

[1] More recent approaches based on predictive coding reconceptualize the cognitive view on sensorimotor behaviour and question the need for having an IM at all, arguing that instead, a single, integrative forward model is sufficient [1]. In our work, we adhere to the standard, two-models view.

Section 3.2, in our case, the IM is much easier to estimate since it is used for relating the pairs of consecutive states by mediating actions.

As mentioned in Introduction, causal cognition in robots has been proposed to include a range of categories, varying in terms of complexity [7]. Here, we focus on the lower end of this spectrum and discuss low-level causality regarding two categories: sensorimotor self-learning (C1) and learning the consequences of one's own actions on objects in the environment (C2).

Our work was motivated by [10], where the authors present CREST, an approach for causal reasoning in simulation to learn the relevant state space for a robot manipulation policy. In their approach, they conduct interventions using internal models (with simplified assumptions) that elicit the structure between the state and action spaces, enabling the construction of neural network policies with only relevant state features as input. They have shown on two representative manipulation tasks (block stacking and crate opening) that the policies were more robust to domain shifts, more sample efficient to learn, and scaled to more complex settings with larger state spaces. CREST was presented as one approach to a broader methodology of structure-based transfer learning from simulation as a new paradigm for sim-to-real robot learning, i.e., structural sim-to-real. The CREST achieves dimensionality reduction for reinforcement learning in the state space; the related work demonstrates that in the action space [14].

3 Methods

3.1 Data Generation

We collected sensorimotor data in a simulated environment using myGym toolkit [20]. In each step, the agent (robotic arm) executes a randomly selected action and observes a new state. Motor babbling is a natural process observed in infants during their first months. In the case of interaction with objects, the concept of intuitive physics becomes relevant. In the case of C1, the arm performs motor babbling and records its joint configuration and Cartesian effector position before and after the execution of an action.

In the case of C2, an object is added to the table in the simulated environment, and the arm has the possibility to interact with it using constrained motor babbling. During an episode, the agent observes potential changes in position, rotation and other defined features of the object, arm and environment in response to the arm's actions.

3.2 Forward and Inverse Models

The generated data is used for learning of the forward and inverse models. The FM is implemented by a feed-forward neural network that learns the mapping

$$\text{FM:}\ [\boldsymbol{s}(t), \boldsymbol{a}(t)] \mapsto \boldsymbol{s}(t+1), \tag{1}$$

where the state vector features $\boldsymbol{s}(t)$ are task-dependent. In our experiments, $\boldsymbol{\theta}(t), \boldsymbol{ef}(t) \subset \boldsymbol{s}(t)$, where $\boldsymbol{\theta}(t)$ is the joint configuration and $\boldsymbol{ef}(t)$ is the effector

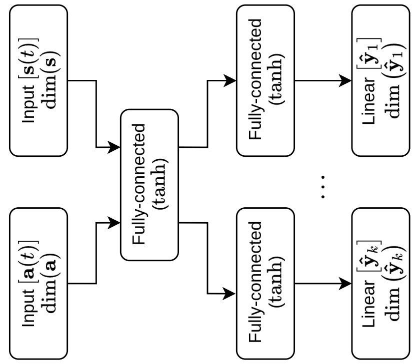

Fig. 1. General forward model architecture.

position in Cartesian 3D space. Here, we assume the robot perception $\boldsymbol{s}(t)$ is reliable, serving as ground truth the agent learns to estimate.[2] We also assume that the action vector is represented as $\boldsymbol{a}(t) = \boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)$, making our approach biologically plausible since the action depends on a current state.[3]

Since the state features can be diverse, the architecture of the FM (Figure 1) contains separate output heads for different state subvectors $\hat{\boldsymbol{y}}_i \subseteq \boldsymbol{s}(t{+}1)$. During the training, loss is computed separately for each subvector, and the model is optimized according to an equally weighted sum of partial losses.

Similarly, the IM is implemented by a feed-forward neural network that learns the mapping

$$\mathrm{IM}\colon\ [\boldsymbol{s}(t), \boldsymbol{s}(t+1)] \mapsto \boldsymbol{a}(t). \tag{2}$$

During the offline training from the generated dataset, we can assume the availability of $\boldsymbol{\theta}(t+1) \subset \boldsymbol{s}(t+1)$, but not during the inference. For this reason, we propose two approaches to the construction of inverse models. The monolithic approach (Figure 2) consists of one neural network taking $\boldsymbol{s}(t)$ and $\boldsymbol{s}'(t+1) = \boldsymbol{s}(t+1) \setminus \boldsymbol{\theta}(t+1)$ (i.e., the original state vector without $\boldsymbol{\theta}$ subvector) as input during both training and inference.

Since learning of the IM mapping (Eq. 2) is less difficult with $\boldsymbol{\theta}(t{+}1)$ available, the second approach (Figure 3) relies on the composition of the base model learning such mapping and the pre-network learning the mapping $[\boldsymbol{s}(t), \boldsymbol{s}'(t+1)] \mapsto \boldsymbol{\theta}(t+1)$ on the generated and pre-computing the approximate value of $\boldsymbol{\theta}(t+1)$ during the inference. This output is then concatenated with the rest of the inputs and fed into the base model. We compare both approaches in Section 4.1.

[2] It should be acknowledged that this reliability assumption holds well in simulation but may not in real robots, when the perception may be inaccurate or even fail.

[3] A commonly used alternative in robotics is to represent an action as a corresponding target joint vector.

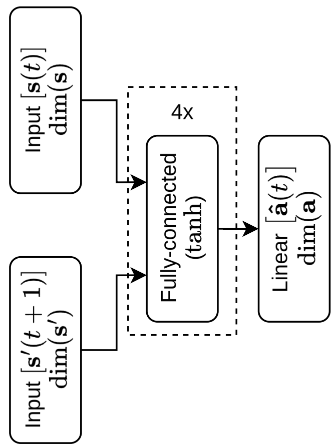

Fig. 2. General monolithic inverse model architecture.

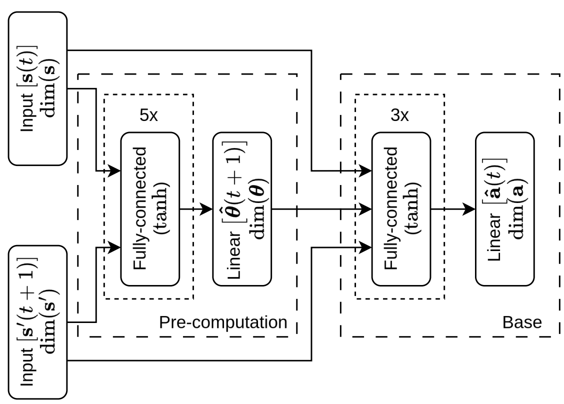

Fig. 3. Inverse model architecture with $\boldsymbol{\theta}(t+1)$ pre-computation pre-network.

3.3 Knowledge Extraction

Trained FM can be analyzed by extracting information about the original environment and a learning session. Our primary focus is on analyzing feature importance, which allows us to highlight state features that cannot be manipulated by the agent actions and thus can be removed, hence reducing the dimensionality of the state space for the specific task and environment. Recent related work by Lee et al. [10], which served as an inspiration, focused on determining relevant state features by conducting intervention on one feature at a time and testing whether the same policy execution led to successful task completion or not. This way, causal dependencies were found.

On the contrary, we do not study causality by direct interaction with an environment but by using trained causal models as proxies containing this information. Using the learned FM, we can determine the relevance of state features in relation to action features by analyzing their importance.

Explaining the behaviour of trained deep neural networks is hard since these networks derive their decisions using a large number of elementary operations. Various approaches have been proposed [5], and one category that we focus on here is based on saliency mapping. Model predictions are typically based on attributions to several features (inputs). SHAP framework [13] unifies different additive feature attribution methods. Here, we were experimenting with Kernel SHAP and Deep SHAP. While Kernel SHAP is model-agnostic and utilizes Linear LIME [17], Deep SHAP is applicable only to neural models and uses attribution rules of DeepLIFT method [19]. Both methods try to approximate the Shapley value of each input feature in relation to an output feature. Shapley value represents an input feature contribution to the output feature prediction. In the context of our work, we can use Shapley values to determine the contribution of each action variable to each state variable for a specific prediction.

In our experiments (Section 4.2), we use Deep SHAP method only, as Kernel SHAP method is significantly slower due to it making no assumptions about the analyzed model. While SHAP methods are local, providing an explanation for one prediction, thanks to their properties, these local explanations can be aggregated across the set of instances, providing us with global feature importance within the analyzed model visualizable using heat maps (e.g., Figure 5).

Additionally, we can use partial dependence plots (PDPs) [3] to visualize the distribution of the contribution of an action feature to a state feature across the set of instances (e.g., Figure 6). From PDPs, we can determine whether there is any relationship between the selected action and the state features and properties of this relationship. High correlation indicates a strong impact of the action feature on the state feature (however, such a relationship should not always be regarded as causal [2]).

4 Experiments

We applied the described methods in two experiments related to categories C1 and C2 of learning causal relations [7]. The structure of both experiments is similar. The first step consists of data generation in a simulated robotic environment (Section 3.1) provided by myGym toolkit [20]. The generated data is subsequently used for training forward and inverse models (Section 3.2), which are further analyzed to reveal causal relations.

4.1 Learning Kinematics

In experiment 1, which focused on sensorimotor learning, we used Franka Emika Panda robotic arm with a gripper and 7 DoF that performed motor babbling.

Environment The simulation ran in 500,000 steps. In each step, a joint configuration $\boldsymbol{\theta} \in \mathbb{R}^8$ is sampled from the normal distribution with limits according to Table 1. The gripper (θ_8) is fixed during the simulation. A motor command

Table 1. Joint motion range in Panda arm used for Experiment 1.

	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8
$q_{\min}$ [rad]	-2.967	-1.833	-2.967	-3.142	-2.967	-0.087	-2.967	0.0
$q_{\max}$ [rad]	2.967	1.833	2.967	0.0	2.967	3.822	2.967	0.0

is executed in 10 substeps before proceeding to the next simulation step, allowing a longer execution time, resulting in the actual action being more similar to the planned one. After the action execution, only the resulting configuration and Cartesian effector position $\boldsymbol{ef} = [ef_x, ef_y, ef_z]$ is recorded, both composing state vector $\boldsymbol{s}(t)$.

Models The FM for this experiment uses two separate output heads, one for joint configuration prediction and the other for effector position prediction. Each head computes a separate mean squared error used as a loss. The model is trained according to the overall loss calculated as a sum of equally weighted head losses. For training the FM, we used Adam optimizer [8] with an initial learning rate $\eta = 10^{-3}$ for 60 epochs. The model was evaluated using 5-fold cross-validation with average MAE for the effector position and joint configuration outputs of 2 mm and 1.2×10^{-3} rad, respectively.

For the IM, we tried both architectural approaches proposed in Section 3.2. All models of both approaches use MSE as a loss function. The pre-computation approach uses a base IM trained first. We experimented with two variants of this model, differing in the unit's activation function at the hidden layer: hyperbolic tangent or ReLU. However, the differences in resulting performance were insignificant.

We also tested the base IM by inputting $\boldsymbol{\theta}(t+1) = \mathbf{0}$ or sampling $\theta_i(t+1)$ from the dataset as an alternative to approximating $\boldsymbol{\theta}(t+1)$ by the pre-network. Sampling from the dataset and inputting zero vector for $\boldsymbol{\theta}(t+1)$ resulted in an MAE of 0.208 rad and 0.457 rad, respectively.

Next, we trained the feature-generating pre-network separately. After the training, the pre-network was put in front of the base model, forming the assembly used for inference. The training hyperparameters and final results of both approaches are shown in Table 2.

Mental simulation We subjected the FM to the chained inference, which consisted of repeatedly querying the model on the previously generated state and random action. Each prediction is then compared with the ground-truth trajectory generated in simulation without the model being given real reference. This test was performed on 30,000 generated trajectories, resulting in linear growth of both the joint and effector position errors (Figure 4).

4.2 Simple Intuitive Physics

This experiment has the purpose of evaluating the proposed methods on the category 2 causal skill proposed by Hellström [7], described as "Learning about how the robot affects the world".

Table 2. Hyperparameters and resulting MAE of approaches to inverse model construction for kinematics data. Results were obtained using 5-fold cross-validation. η and λ denote an initial learning rate and initial weight decay, respectively, of Adam [8] and AdamW [12] optimizers.

Approach	Model	Epochs	Optimizer	MAE [rad]
Monolithic	Base	1,000	AdamW($\eta = 10^{-3}$, $\lambda = 0.004$)	0.0120
Pre-computation	Base	100	Adam($\eta = 10^{-3}$)	9.594×10^{-4}
	Pre-network	4,000	AdamW($\eta = 10^{-3}$, $\lambda = 0.004$)	0.0126
	Assembly	N/A	N/A	0.0139

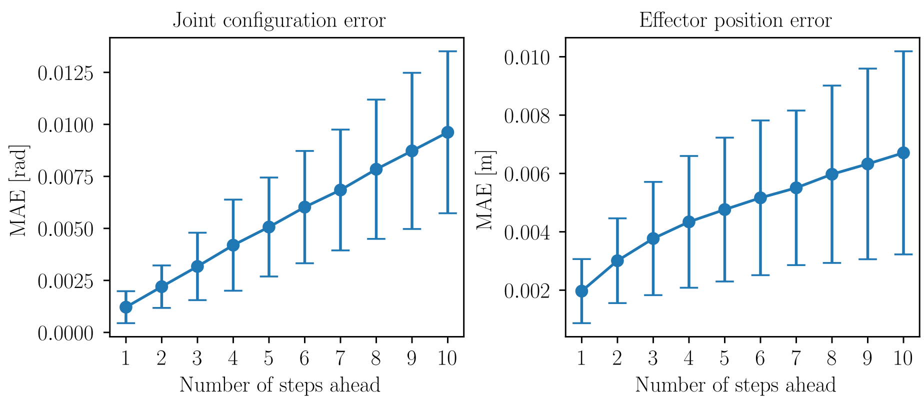

Fig. 4. Error of the forward model during mental simulation 10 steps ahead.

Environment The experiment utilizes the KUKA LBR iiwa robotic arm with a magnetic endpoint as an agent. The task consisted of the arm randomly switching the magnet. If it was not holding anything, the arm navigated to the magnetized cube lying on the table, picked it up, and randomly manoeuvred with it in the space for a random duration. After that, the magnet was turned off, the cube was released, and the arm babbled empty-handed for a random duration.

The goal of this experiment was to let the agent learn the simple physics of the cube as well as its kinematics. Knowledge gained by the agent in this experiment can be thus understood as a superset of knowledge from the kinematics experiment (Section 4.1). Additionally, we wanted to verify whether the model architectures specified in Section 3.2 would efficiently work with state spaces of higher dimensionality.

The data-generating simulation ran in 4,000 episodes, lasting 500 iterations each. After each iteration, we recorded the final joint configuration $\boldsymbol{\theta} \in \mathbb{R}^7$, effector position and rotation $\boldsymbol{ef} = \left[ef_x, ef_y, ef_z, ef_{rx}, ef_{ry} ef_{rz}\right]$, object information (its position, rotation and color) $\boldsymbol{o} = [o_x, o_y, o_z, o_{rx}, o_{ry}, o_{rz}, o_R, o_G, o_B]$, and the magnet state *mgt*. Object colour features were added as control variables, randomized at the start of each episode, and did not change during it.

Models Same as in the previous experiment, we trained an FM and a monolithic IM on the generated data. The FM uses separate output heads for object position, object rotation, colour, joint configuration, effector position and rotation prediction. Each head computes a separate MSE, which is used as a loss. The FM was trained for 100 epochs using Adam optimizer [8] with the initial learning rate $\eta = 10^{-3}$. For the final results of this model, see Table 3.

Table 3. Errors of respective output heads of the forward model for physics data. Results were obtained using 5-fold cross-validation.

Output head	MAE	Output head	MAE
Object position	0.0089 m	Effector position	0.008 m
Object rotation	0.0721 rad	Effector rotation	0.0625 rad
Object color	0.004	Joint configuration	0.0084 rad
		Magnet state	1.3×10^{-4}

In this experiment, we applied only a monolithic approach to the inverse model construction. The model is optimized according to separate MSE for joint and magnet action prediction. The training was facilitated by AdamW optimizer [12] with the initial learning rate $\eta = 10^{-3}$ and initial weight decay $\lambda = 0.004$ for 1,000 epochs with the final MAE of joint action prediction 0.0077 rad and of magnet action prediction 4.56×10^{-4}.

Knowledge extraction The trained forward model is further analyzed using methods proposed in Section 3.3. The analysis was performed using a sample of 200 observations from the generated dataset.

The resulting global contribution heat map generated for this experiment is shown in Figure 5. The y-axis denotes the action of each joint and a magnetic endpoint of the arm. The x-axis contains defined environment state features. The colour of each square corresponds to the magnitude of contributions of action features to the state features averaged across the selected sample.

The figure shows, for instance, that joint 6 is not used in the sampled observation data. In addition, the colour of the object is irrelevant in this case as no action can affect it and thus could be removed (or ignored) from the state space. On the other hand, all action features, except a_6, affect most object features. This low-level knowledge can be useful for causal analysis at higher levels.

Feature importance and dependencies can also be studied using feature contribution distributions and partial dependence plots. Figure 6 shows a sample of PDPs generated from feature contribution data output by Deep SHAP method [13]. Averaged absolute contribution across these distributions corresponds to the values in Figure 5.

From the presented sample of PDPs, we can observe that action feature a_0 (movement of joint 0) has a substantial impact on state features θ_0 (state of

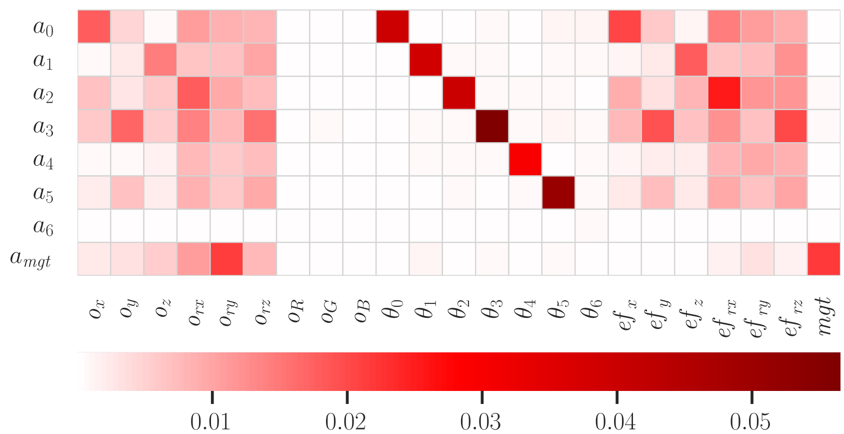

Fig. 5. Contribution heat map generated by Deep SHAP method on the forward model showing magnitude of contribution of specific actions to output features.

joint 0) and o_x (object position on the x-axis). Moreover, action feature a_2 does not significantly correlate with the contribution to state feature o_B (blue color component of the object) with contribution centered around 0.0 indicating a_2 does not profoundly impact o_B. Last, the action of the magnetic endpoint a_{mgt} is prevalently null as the state of the magnet does not often change between iterations. However, when it does, it significantly contributes to o_z (object position on the z-axis) since turning the magnet on ($a_{mgt} = 1$) or off ($a_{mgt} = -1$) in this experiment is followed by lifting the object in the air or dropping it on the table.

5 Conclusion

In this paper, we explored causal relations by learning the forward and inverse models on synthetic data generated in simulation. We confirmed that the forward model constructed using the proposed approach can be successfully used for mental simulation, possibly helping with action planning by predicting future states based on causality observed in the data. Additionally, we explored approaches to inverse model construction allowing the model to predict the action needed for transition between subsequent states.

Moreover, we demonstrated the capability of extracting knowledge about the behaviour of the environment from the trained forward model using the explainability methods. We proposed that information obtained this way can be used to determine relevant state feature, serving as a basis for dimensional reduction. Our method can be applied to scenarios that are much more complex and much harder for humans to understand, and thus, it can be an important tool for extracting causal knowledge. For future work, we plan to investigate the task of action planning as an imitation learning assisted by the proposed models.

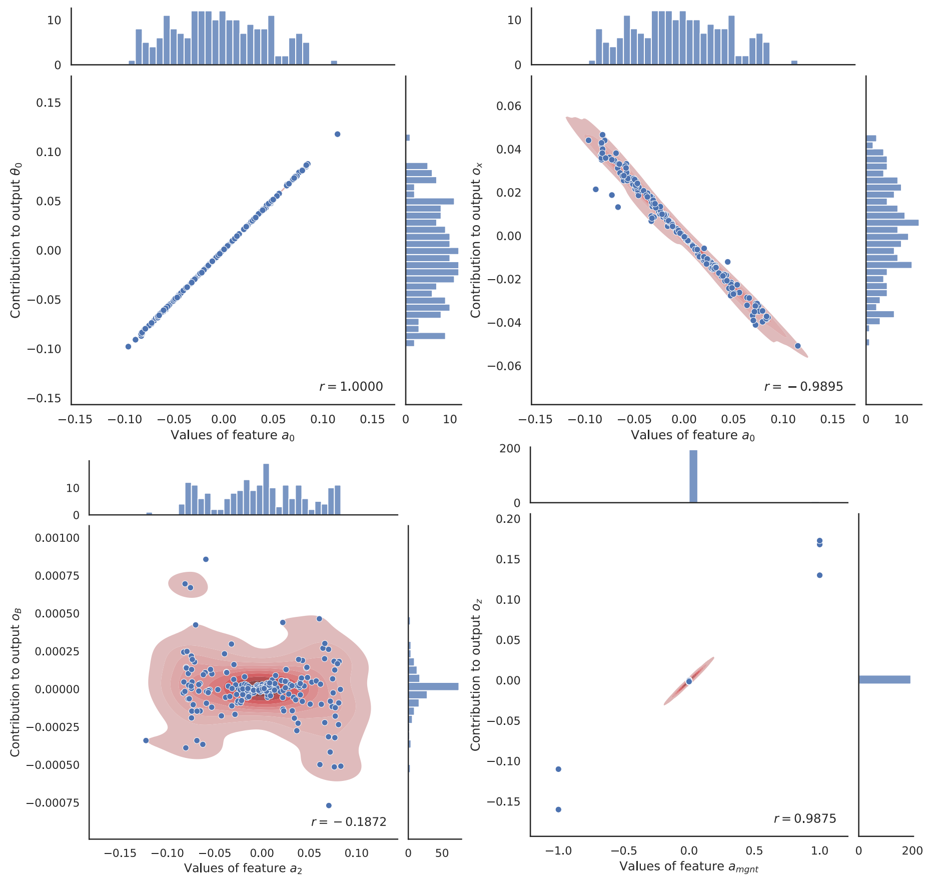

Fig. 6. A sample of partial dependence plots generated by Deep SHAP method applied to the forward model showing correlation between a value of a specific action component and its contribution to an output variable.

References

1. Ciria, A., Schillaci, G., Pezzulo, G., Hafner, V.V., Lara, B.: Predictive Processing in Cognitive Robotics: A Review. Neural Computation **33**(5), 1402–1432 (2021). https://doi.org/10.1162/neco_a_01383
2. Dillon, E., LaRiviere, J., Lundberg, S., Roth, J., Syrgkanis, V.: Be careful when interpreting predictive models in search of causal insights (2021), https://towardsdatascience.com/be-careful-when-interpreting-predictive-models-in-search-of-causal-insights-e68626e664b6
3. Friedman, J.H.: Greedy Function Approximation: A Gradient Boosting Machine. The Annals of Statistics **29**(5), 1189–1232 (2001). https://doi.org/10.1214/aos/1013203451

4. Gerstenberg, T., Tenenbaum, J.B.: Intuitive theories. In: Waldmann, M.R. (ed.) The Oxford Handbook of Causal Reasoning, pp. 515–548. Oxford University Press (2017). https://doi.org/10.1093/oxfordhb/9780199399550.013.28
5. Gilpin, L.H., Bau, D., Yuan, B.Z., Bajwa, A., Specter, M., Kagal, L.: Explaining Explanations: An Overview of Interpretability of Machine Learning. In: 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA). pp. 80–89. IEEE (2018). https://doi.org/10.1109/dsaa.2018.00018
6. Gärdenfors, P., Lombard, M.: Causal cognition, force dynamics and early hunting technologies. Frontiers in Psychology **9** (2018). https://doi.org/10.3389/fpsyg.2018.00087
7. Hellström, T.: The relevance of causation in robotics: A review, categorization, and analysis. Paladyn, Journal of Behavioral Robotics **12**(1), 238–255 (2021). https://doi.org/10.1515/pjbr-2021-0017
8. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization (2014). https://doi.org/10.48550/ARXIV.1412.6980
9. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. Behavioral and Brain Sciences **40** (2016). https://doi.org/10.1017/s0140525x16001837
10. Lee, T.E., Zhao, J.A., Sawhney, A.S., Girdhar, S., Kroemer, O.: Causal reasoning in simulation for structure and transfer learning of robot manipulation policies. In: IEEE International Conference on Robotics and Automation (ICRA). IEEE (2021). https://doi.org/10.1109/icra48506.2021.9561439
11. Lombard, M., Gärdenfors, P.: Tracking the evolution of causal cognition in humans. Journal of Anthropological Sciences **95**, 219–234 (2017). https://doi.org/10.4436/JASS.95006
12. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2017). https://doi.org/10.48550/arXiv.1711.05101
13. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17, vol. 30, p. 4768–4777 (2017)
14. Nouri, A., Littman, M.L.: Dimension reduction and its application to model-based exploration in continuous spaces. Machine Learning **81**(1), 85–98 (2010). https://doi.org/10.1007/s10994-010-5202-y
15. Pearl, J.: Causality: Models, Reasoning, and Inference. Cambridge University Press, New York, NY, 2 edn. (2009). https://doi.org/10.1017/cbo9780511803161
16. Peters, J., Janzing, D., Bernard, S.: Elements of Causal Inference - Foundations and Learning Algorithms. MIT Press, Cambridge, MA, USA (2017)
17. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 1135–1144. ACM (2016). https://doi.org/10.1145/2939672.2939778
18. Schölkopf, B.: Causality for Machine Learning, pp. 765–804. Association for Computing Machinery, New York, NY, USA, 1 edn. (2022). https://doi.org/10.1145/3501714.3501755
19. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 3145–3153. PMLR (2017), https://proceedings.mlr.press/v70/shrikumar17a.html

20. Vavrečka, M., Sokovnin, N., Mejdrechová, M., Šejnová, G.: MyGym: Modular Toolkit for Visuomotor Robotic Tasks. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI). pp. 279–283. IEEE (2021). https://doi.org/10.1109/ictai52525.2021.00046
21. Wolpert, D.M., Kawato, M.: Multiple paired forward and inverse models for motor control. Neural Networks **11**(7–8), 1317–1329 (1998). https://doi.org/10.1016/s0893-6080(98)00066-5
22. Zhang, K., Schölkopf, B., Spirtes, P., Glymour, C.: Learning Causality and Causality-Related Learning: Some Recent Progress. National Science Review **5**(1), 26–29 (2017). https://doi.org/10.1093/nsr/nwx137
23. Zhu, Y., Gao, T., Fan, L., Huang, S., Edmonds, M., Liu, H., Gao, F., Zhang, C., Qi, S., Wu, Y.N., Tenenbaum, J.B., Zhu, S.C.: Dark, beyond deep: A paradigm shift to cognitive AI with humanlike common sense. Engineering **6**(3), 310–345 (2020). https://doi.org/10.1016/j.eng.2020.01.011