CAUSALWORLD: A ROBOTIC MANIPULATION BENCHMARK FOR CAUSAL STRUCTURE AND TRANSFER LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite recent successes of reinforcement learning (RL), it remains a challenge for agents to transfer learned skills to related environments. To facilitate research addressing this, we propose *CausalWorld*, a benchmark for causal structure and transfer learning in a robotic manipulation environment. The environment is a simulation of an open-source robotic platform, hence offering the possibility of sim-to-real transfer. Tasks consist of constructing 3D shapes from a given set of blocks - inspired by how children learn to build complex structures. The key strength of *CausalWorld* is that it provides a combinatorial family of such tasks with common causal structure and underlying factors (including, e.g., robot and object masses, colors, sizes). The user (or the agent) may intervene on all causal variables, which allows for fine-grained control over how similar different tasks (or task distributions) are. One can thus easily define training and evaluation distributions of a desired difficulty level, targeting a specific form of generalization (e.g., only changes in appearance or object mass). Further, this common parametrization facilitates defining curricula by interpolating between an initial and a target task. While users may define their own task distributions, we present eight meaningful distributions as concrete benchmarks, ranging from simple to very challenging, all of which require long-horizon planning and precise low-level motor control at the same time. Finally, we provide baseline results for a subset of these tasks on distinct training curricula and corresponding evaluation protocols, verifying the feasibility of the tasks in this benchmark.[1]

1 INTRODUCTION

Benchmarks have played a crucial role in advancing entire research fields, for instance computer vision with the introduction of CIFAR-10 and ImageNet (Krizhevsky et al., 2009; 2012). When it comes to the field of reinforcement learning (RL), similar breakthroughs have been achieved in domains such as game playing (Mnih et al., 2013; Silver et al., 2017), learning motor control for high-dimensional simulated robots (Akkaya et al., 2019), multi-agent settings (Baker et al., 2019; Berner et al., 2019) and for studying transfer in the context of meta-learning (Yu et al., 2019). Nevertheless, trained agents often fail to transfer the knowledge about the learned skills from a training environment to a different but related environment sharing part of the underlying task structure. This can be attributed to the fact that it is quite common to evaluate an agent on the training environments themselves, which leads to overfitting on these narrowly defined environments (Whiteson et al., 2011), or that algorithms are compared using highly engineered and biased reward functions which may result into learning suboptimal policies with respect to the desired behaviour; this is particularly evident in robotics.

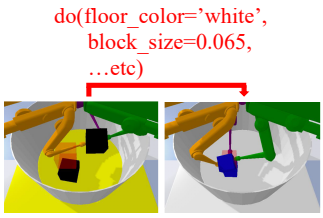

Figure 1: Example of do-interventions on exposed variables in `CausalWorld`.

[1] https://sites.google.com/view/causalworld-iclr/home

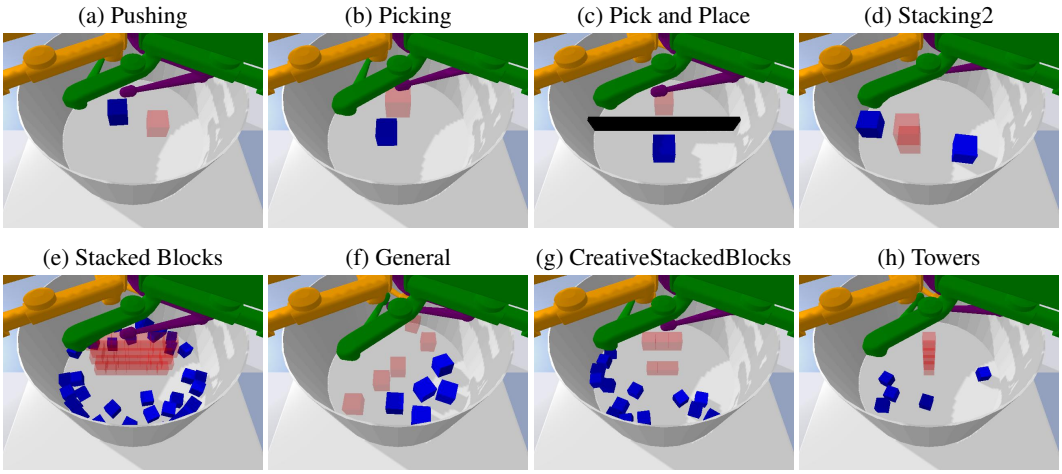

Figure 2: Example tasks from the task generators provided in the benchmark. The goal shape is visualized in opaque red and the blocks in blue.

In existing benchmarks (Yu et al., 2019; Goyal et al., 2019; Cobbe et al., 2018; Bellemare et al., 2013; James et al., 2020) the amount of shared causal structure between the different environments is mostly unknown, e.g. in the Atari Arcade Learning environments, it is unclear how to quantify the underlying similarities between different Atari games and we generally do not know to which degree an agent can be expected to generalize. To overcome these limitations, there is a need for a benchmark with a diverse set of environments which, in contrast to previous designs, should share a large set of parameters and parts of the causal structure. Being able to intervene on these parameters (individually or collectively) allows the experimenter to evaluate agents' generalization ability with respect to different types and extents of changes in the environment. These parameters can be varied gradually, which yields a continuum of similar environments that allows for fine-grained control of training and test distributions. The objective of this work is to address these points by introducing a novel benchmark in a robotic manipulation environment which we call `CausalWorld`.

A remarkable skill that humans learn to master relatively early on in their life is building complex structures using their spatial-reasoning and dexterous manipulation abilities (Casey et al., 2008; Caldera et al., 1999; Kamii et al., 2004). Playing with toy blocks constitutes a natural environment for children to develop important critical and visual-spatial skills, helping them 'generalize' in building complex composition designs from presented or imagined goal structures (Verdine et al., 2017; Nath & Szücs, 2014; Dewar, 2018; Richardson et al., 2014). Inspired by this, `CausalWorld` is designed to aid in learning and investigating these skills in a corresponding simulated robotics manipulation environment of the open-source TriFinger robot platform from Wüthrich et al. (2020), which can be built in the real world. Tasks are formulated as building 3D goal shapes using a set of available blocks by manipulating them - as seen in Fig. 1. This task definition yields a combinatorially large and diverse space of target structures. We subsequently provide eight distinct task generators which generate different environments covering distinct target shape families of interest. The resulting environment instances range from simple to extremely challenging. Additionally, by using a common goal formulation, we can define a success metric which is shared across all environments: We measure the volumetric overlap of the objects with the goal shape.

`CausalWorld` improves upon previous benchmarks by exposing a large set of parameters in the causal generative model of the environments, such as properties of blocks, goals, robot links and others like gravity. The possibility to intervene on any of these environment properties at any point in time allows one to set up training environments in a curriculum manner and evaluate agents across different generalization axes using a broad set of evaluation protocols. Furthermore, emphasizing the real-world relevance of this benchmark as opposed to earlier ones (Chevalier-Boisvert et al., 2018; Cobbe et al., 2018), researchers may build their own real-world platform of this simulator at low cost, as detailed in Wüthrich et al. (2020), and transfer their trained policies to the real world.

Benchmark	do-interventions interface	procedurally generated environments	online distribution of tasks	setup custom curricula	disentangle generalization ability	real-world similarity	open-source robot	low-level motor control	long-term planning	unified success metric
RLBench	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
MetaWorld	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗
IKEA	✗	✗	✗	✗	✗	✓	✗	✓	✓	✓
MuJoBan	✗	✗	✗	✓	✗	✓	✗	✓	✓	✓
BabyAI	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓
CoinRun	✗	✓	✗	✗	✗	✗	✗	✗	✗	✓
AtariArcade	✗	✗	✗	✗	✗	✗	✗	✗	✓/✗	✓
`CausalWorld`	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of Causal World with RLBench (James et al., 2020), MetaWorld (Yu et al., 2019), IKEA (Lee et al., 2019), BabyAI (Chevalier-Boisvert et al., 2018), CoinRun (Cobbe et al., 2018), AtariArcade (Bellemare et al., 2013), MuJoBan etc. (Mirza et al., 2020),

Finally, by releasing this benchmark we hope to facilitate research in causal structure learning, i.e. learning the causal graph or certain aspects of it, as we operate in a complex real-world environment whose dynamics follow the laws of physics which induce causal relations between the variables. Changes to the variables we expose can be considered do-interventions on the underlying structural causal model (SCM). Consequently, we believe that this benchmark offers an exciting opportunity to investigate causality and its connection to RL and robotics.

Our main contributions can be summarized as follows:

- We propose `CausalWorld`, a new benchmark comprising a parametrized family of robotic manipulation environments for advancing out-of-distribution generalization and causal structure learning in RL.
- We provide a *systematic* way of defining curricula and disentangling generalization abilities of trained agents by allowing do-interventions to be performed on all environment variables (parameters and states).
- We establish baseline results for some of the available tasks under different learning algorithms, thus verifying the feasibility of the tasks.
- We show how different learning curricula affect generalization across different axes by reporting some of the in-distribution and out-of-distribution generalization capabilities of the trained agents.

2 CAUSALWORLD BENCHMARK

Here, we make the desiderata outlined in the introduction more precise:

1. The set of environments should be sufficiently diverse to design challenging transfer tasks.
2. We need to be able to intervene on different properties (e.g. masses, colors) individually, such that we can investigate different types of generalization.
3. It should be possible to convert any environment to any other environment by gradually changing its properties through interventions; this requirement is important for evaluating different levels of transfer and for defining curricula.
4. The environments should share some causal structure to allow algorithms to transfer the learned causal knowledge from one environment to another.
5. There should be a unified measure of success, such that an objective comparison can be made between different learning algorithms.
6. The benchmark should make it easy for users to define meaningful distributions of environments for training and evaluation. In particular, it should facilitate evaluation of in-distribution and out-of-distribution performance.
7. The simulated benchmark should have real-world counterpart to allow for sim2real.

In light of these desiderata, we propose a setup in which a robot must build goal shapes using a set of available objects. It is worth noting that similar setups were proposed previously in a less

realistic setting as in (Janner et al., 2018; Bapst et al., 2019; McCarthy et al.; Akkaya et al., 2019; Fahlman, 1974; Winston, 1970; Winograd, 1972). Specifically, a task is formulated as follows: *given a set of available objects with different properties (such as size, mass, friction, etc.) the agent needs to build a specific goal structure*, see Fig. 1 for an example. The vast amount of possible target shapes and environment properties makes this a diverse and challenging setting to evaluate different generalization aspects. `CausalWorld` is a simulated version (using the Bullet physics engine (Coumans et al., 2013)) of the open-source TriFinger robot platform from Wüthrich et al. (2020). Each environment is completely defined by a set of variables such as gravity, floor friction, stage color, floor color, joint positions, various block parameters (e.g. size, color, mass, position, orientation), link colors, link masses and the goal shape. See Table 3 in the Appendix for a subset of these variables.

Desideratum 1 is satisfied since different environment properties and goal shapes give rise to very different tasks, ranging from relatively easy (e.g. re-positioning a single cube) to extremely hard (e.g. building a complex multi-level goal shape). Desideratum 2 is satisfied because we allow for arbitrary interventions on these properties, hence users or agents may change parameters individually or jointly. Desideratum 3 is satisfied because the parameters can be changed gradually. Desideratum 4 is satisfied because all the environments share the causal structure of the robot, and one may also use subsets of environments which share even more causal structure. We satisfy desideratum 5 by defining the measure of success for all environments as the volumetric overlap of the goal shape with available objects. Further, by splitting the set of parameters into a set A, intended for training and in-distribution evaluation, and a set B, intended for out-of-distribution evaluation, we satisfy desideratum 6. Finally, since the TriFinger robot (Wüthrich et al., 2020) can be built in the real-world, we satisfy desideratum 7. Desideratum 7 and 2 are in partial conflict since sim2real is only possible for the tasks which are constrained to the variables on which the robot can physically act upon.

Task generators: The space of feasible goal shapes one can match with a given set of objects is combinatorially exponential. However, only a small subset of these goal shapes comprise topologically meaningful shapes. To cover collections spanning a family of similar goal shapes, `CausalWorld` allows for defining task generators which can procedurally generate a variety of different goal shapes in an environment. Each task generator is initialized with a default goal shape from its corresponding family and comes with a sampler to sample new goal shapes from the same family. Additionally, upon construction, one can specify the environments' initial state and initial goal shape structure when deviating from the default. The maximum episode time to build a given shape is `number_of_blocks` $\times$ 10 seconds. Initially, `CausalWorld` contains eight task generators that could be used to sample goal shapes from different goal shape families (see Fig. 2).

- Three generators are constrained to a goal shape with a single block: *Pushing* with the goal shape on the floor, *Picking* having the goal shape defined above the floor and *Pick and Place* where a fixed obstacle is placed between the initial block and goal pose.
- *Stacking2* involves a goal shape of two stacked blocks, which can also be considered one instance of the *towers* generator.
- The remaining generators use a variable number of blocks to generate much more complex and challenging target shapes with details in the appendix: *Towers*, *Stacked Blocks*, *Creative Stacked Blocks* and *General*.

Given that building new environments using current physics simulators is usually tedious, we provide a simple API for users who wish to create new task generators, for new challenging shape families which may be added to `CausalWorld's` task generators repository.

Action and Observation Spaces: The robot's action space $A \in \mathbb{R}^9$ can be chosen to operate in either joint position control mode, joint torque control mode, end-effector position control mode, or the delta of each. To address the different challenges in using high-dimensional visual observations as well as using a structured representation, we provide two observation modes: *structured* as well as *pixel*. In the *structured* mode, the low-dimensional observation vector o follows a common rule for the ordering of the relevant variables, such as joints position, joints velocity, blocks linear velocity, time left for the task..etc. Thus, the observation space size depends on the number of blocks, which could potentially change with every new goal sampled, e.g. in Towers, (Creative) Stacked

Blocks and General; therefore $O \in \mathbb{R}^{d_o}$ where d_o varies across different environments. On the contrary, in the *pixel* mode, the agent receives six different RGB images, where $O \in \mathbb{R}^{6\times3\times128\times128}$, the first three images are rendered from different cameras mounted around the TriFinger robot, and the last three images specify the goal image of the target shape rendered from the same cameras. This mode can be mirrored on the real robotic platform and aids in investigating object-based learning approaches from pixel data as well as learning visual goal conditioned policies. Additionally, `CausalWorld` allows for setting up a fully customized observation space, if needed.

Rewards: The reward function r is defined uniformly across all possible goal shapes as the fractional volumetric overlap of the blocks with the goal shape, which ranges between 0 (no overlap) and 1 (complete overlap). This shared success metric can be returned at each time step where its scale is independent of the goal shape itself. Thus, an agent that learned this shared reward function r from several different tasks could in principle use it to solve unseen goal structures. There is also the possibility of modifying the reward function to 1) sparsify the reward further by returning a binary reward signal instead, or 2) add a dense reward function in order to introduce inductive biases via domain knowledge and solution guidance. We hope that the considerable complexity and space of goal shapes motivate and accelerate the development of algorithms that are not dependent on highly tuned reward functions anymore by putting more focus on optimal learning curricula instead, which might help to overcome these limitations, and `CausalWorld` provides a convenient way for specifying them via intervention actors.

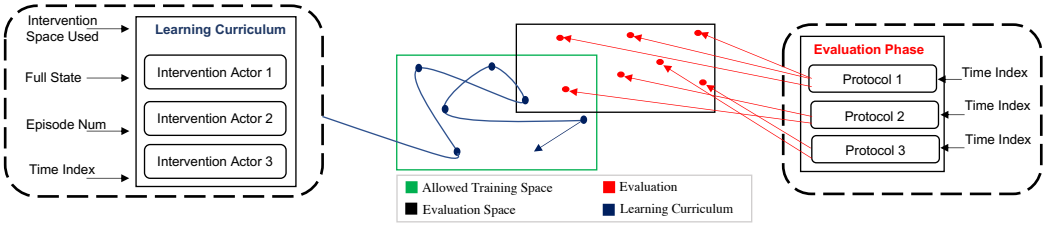

Figure 3: Key components for generic training and evaluation of RL agents. Left: A learning curriculum which is composed of various intervention actors that decide on which variables to intervene on (for a valid intervention, values need to be in the allowed training space (ATS)). Right: Evaluation protocols are shown which may intervene on variables at episode resets or within episodes (for a valid intervention, values need to be in the evaluation space (ES)). Middle: we represent the ATS and ES, where each intervention results in one point in the spaces. As shown ATS and ES may intersect, eg. if the protocols are meant to evaluate in-distribution generalization. A learning curriculum is represented by subsequent interventions navigating the ATS resulting in the corresponding points in the space.

Training and evaluation spaces: In this benchmark, a learning setting consists of an allowed training space (ATS) and an evaluation space (ES), both of which are subspaces of the full parameter space. During training, in the simplest setting, parameters are sampled iid from the ATS. However, unlike existing benchmarks, `CausalWorld` allows for curricula within the ATS as well as settings where the agent itself intervenes on the parameters within an episode (see Fig. 3). Similarly, during evaluation, parameters may be sampled iid from the evaluation space at each episode reset, or there can be interventions within an episode. Moreover, in order to retrieve the setting considered in most RL benchmarks, we could set the ATS and the ES to be identical and intervene only on object and robot states (and keep other environment properties constant) at each episode reset. However, to evaluate out-of-distribution generalization, one should set the two spaces (ATS and ES) to be different; possibly even disjoint. Additionally, to evaluate robustness with respect to a specific parameter (e.g. object mass), one may define the training and evaluation spaces to only differ in that particular parameter. In order to facilitate the definition of appropriate training and evaluation settings, we pre-define two disjoint sets, $\mathbf{A}_i$ and $\mathbf{B}_i$, for each parameter i. Through this, one can for instance define the training space to be $\mathbf{A}_1 \times \mathbf{A}_2 \times ...$ and the evaluation space to be $\mathbf{B}_1 \times \mathbf{B}_2 \times ...$ to assess generalization with respect to all parameters simultaneously. Alternatively, the evaluation space could be defined as $\mathbf{A}_1 \times \mathbf{A}_2 \times ... \times \mathbf{B}_i \times \mathbf{A}_{i+1} \times ...$ to assess generalization with respect to parameter i only. Lastly, users may also define their own spaces which could then be integrated into the benchmark to give rise to new learning settings.

Intervention actors: To provide a convenient way of specifying learning curricula, we introduce intervention actors. At each time step, such an actor takes all the exposed variables of the environment as inputs and may intervene on them. To encourage modularity, one may combine multiple actors in a learning curriculum. This actor is defined by the episode number to start intervening, the episode number to stop intervening, the timestep within the episode it should intervene and the episode periodicity of interventions. We provide a set of predefined intervention actors, including an actor which samples parameters randomly at each episode reset, which corresponds to domain-randomization. It is also easy to define custom intervention actors, we hope that this facilitates investigation into optimal learning curricula (see Fig. 3).

3 Related Work

Previous benchmarks proposed for RL mostly focused on the single task learning setting such as OpenAI Gym and DM control suite (Tassa et al., 2018; Brockman et al., 2016). Although, a recent line of work, e.g. Meta-World and RLBench (Yu et al., 2019; James et al., 2020) aim at studying multi-task learning as well as meta-learning, respective benchmarks mostly exhibit non-parametric hand-designed task variations which makes it ambiguous and not explicit how much structure is shared between them. For instance, it is not clear how different it is to "open a door" compared to "opening a drawer". To address the ambiguity in the shared structure between the tasks, `CausalWorld` was designed to allow interventions to be performed on many environment variables giving rise to a large space of tasks with well-defined relations between them, which we believe is a missing key component to address generalization in RL.

Similar parametric formulations of different environments were used in experiments in the generalization for RL literature, which have played an important role in advancing the field (Packer et al., 2018; Rajeswaran et al., 2017; Pinto et al., 2017; Yu et al., 2017; Henderson et al., 2017a; Dulac-Arnold et al., 2020; Chevalier-Boisvert et al., 2018). In these previous works, variables were mostly assigned randomly as opposed to the full control over the variables in `CausalWorld` by allowing do-interventions.

Another important remaining challenge for the RL community is the standardization of the reported learning curves and results. RL methods have been shown to be sensitive to a range of different factors (Henderson et al., 2017b). Thus it is crucial to devise a set of metrics that measure reliability of RL algorithms and ensure their reproducibility. Chan et al. (2019) distinguishes between several evaluation modes like "evaluation during training" and "evaluation after learning". Osband et al. (2019) recently proposed a benchmarking suite that disentangles the ability of an algorithm to deal with different types of challenges. Its main components are: enforcing a specific methodology for an agent's evaluation beyond the environment definition and isolating core capabilities with targeted 'unit tests' rather than integrating the general learning ability.

4 Experiments

To verify the feasibility of this benchmark, we evaluate current state-of-the-art model-free (MF-RL) algorithms on a subset of the goal shape families described in Section 2: (a) Pushing, (b) Picking, (c) Pick and Place, and (d) Stacking2. We argue that these goal shapes reflect primitive skills that are required to solve far more complex target shape families.

Setup: One objective of this benchmark is to deviate from the widely adopted approach of training and evaluating agents in narrowly defined environments. To emphasize the effect of different training distributions, we set up three handcrafted training curricula:

- Curriculum 0: no environment changes; each episode is initialized from the default task lying in space A - note that here the initial state never changes (i.e. no interventions).
- Curriculum 1: goal shape randomization; at the beginning of each episode a new goal shape is sampled from space $\mathbf{A}$ (i.e. interventions on goal position and orientation).

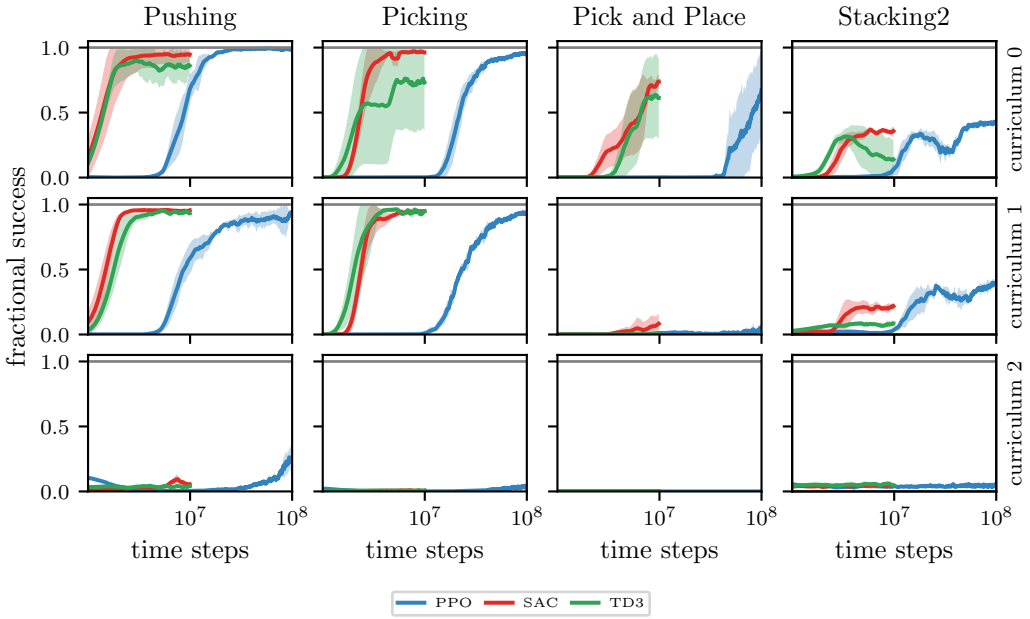

Figure 4: Fractional success curves averaged over five random seeds for the tasks and learning algorithms specified above, under three different training curricula: (0) no curriculum, (1) goal position and orientation randomization in space **A** every episode and (2) a curriculum where we intervene upon all variables in space **A** simultaneously every episode.

- Curriculum 2: full randomization w.r.t. the task variables[2]; every episode a simultaneous intervention on all variables is sampled from space **A** (i.e. can be seen as equivalent to extreme domain randomization in one space).

The curriculum will affect the generalization capabilities of the trained agents. With `CausalWorld`'s formulation, these generalization capabilities can easily be disentangled and benchmarked quantitatively, as explained in Section 2. For each of the goal shape families (a, b, c, d from Fig. 2), we train agents under the three described curricula using the following MF-RL algorithms: The original Proximal Policy Optimization (PPO) from Schulman et al. (2017), Soft Actor-Critic (SAC) from Haarnoja et al. (2018) and the Twin Delayed DDPG (TD3) from Fujimoto et al. (2018). Each task-generator was equipped further with a hand-designed dense reward function as we did not observe any success with the sparse reward only. Each of the mentioned setups is trained for five different random seeds, resulting in 180 trained agents.

Challenging model-free RL methods: We report the training curves averaged over the random seeds in Fig. 4. As can be seen from these fractional success training curves, MF-RL methods are capable of solving the single block goal shapes seen during training time given enough experience. However, we observe that none of the methods studied here managed to solve stacking two blocks. The score below 0.5 indicates that it only learns to push the lower cube into the goal shape. This emphasizes that multi-object target shapes can become nontrivial quickly and that there is a need for methods making use of the modular structure of object-based environments.

To no surprise, the training curriculum has a major effect on learning, but the interpretation of generalization capabilities becomes much more explicit in the following subsection. For example, methods rarely manage to pick up any significant success signal under full extreme domain randomization as in curriculum 2, even after 100 million timesteps. Note that these curves represent the scores under the shapes and conditions of the actual training environments. Therefore, we need the capability to setup different protocols, in other words standardized sets of evaluation environment, that allow to benchmark learned skills of different agents.

[2]Note that each task generator can suppress interventions that would yield goal shapes outside its family

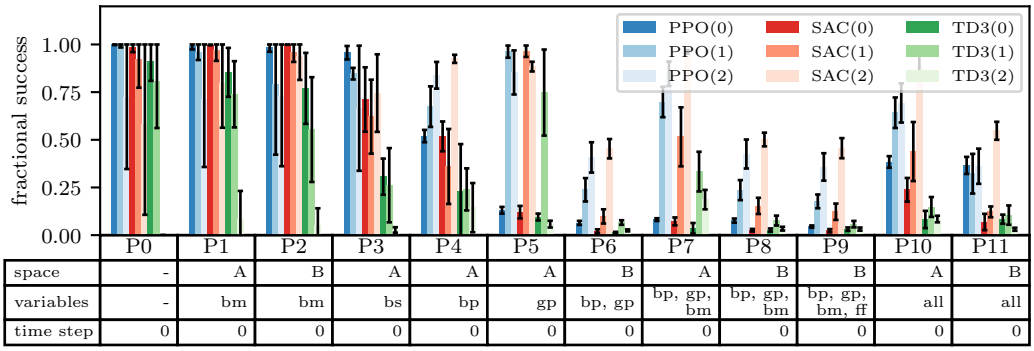

Figure 5: Evaluation scores for pushing baselines. Each protocol was evaluated for 200 episodes and each bar represents the average last fractional success score over five models with different random seeds. The intervention variables listed under each protocol in the "variables" row are sampled randomly from the specified space while all other variables remain fixed [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction]. The interventions are performed on the environment at timestep 0 (i.e. at the begnining of the episode) as shown in the above table.

Benchmarking generalization capabilities along various axes: For each of the four goal shape families, we define a set of 12 evaluation protocols that we consider meaningful and representative for benchmarking the different algorithms. In the protocols presented here, we sample the values from a protocol-specific set of variables at the start of each episode while keeping all other variables fixed to their default values. After evaluating an agent on 200 episodes, we compute the fractional success score at the last time step of each episode and report the mean. These evaluation protocols allow to disentangle generalization abilities, as they show robustness with respect to different types of interventions Fig. 5, The following are some of the observations we made for pushing:

- Agents that were trained on the default pushing task environment (curriculum 0) do well (as expected) on the default task (P0). Interestingly, we likewise see a generalization capability to initial poses from variable space A (P4). This can be explained by a substantial exploration of the block positions via manipulation during training. Similarly, we see that the agents exhibit weaknesses regarding goal poses (P5) but overfit on their training settings instead.
- For agents trained with goal pose randomization (curriculum 1) we see similar results as with curriculum 0, with the difference that agents under this curriculum generalize robustly to different goal poses (P5). This is because it saw many goal poses in space A.
- Finally, agents that experience extreme domain randomization (curriculum 2) at training time, fail to learn any relevant skill as shown by the flat training curve. A possible explanation for this behavior is that the agent might need more data and optimization steps to handle this much more challenging setting. An-

Curriculum	Intervention Variable(s)	Allowed Training Space
0	none	-
1	goal pose	A
2	all	A

Evaluation Protocol	Intervention Variable(s)	Evaluation Space
P0	none	-
P1	block mass	A
P2	block mass	B
P3	block size	A
P4	block pose	A
P5	goal pose	A
P6	block pose goal pose	B
P7	block pose goal pose block mass	A
P8	block pose goal pose block mass	B
P9	block pose goal pose block mass floor friction	B
P10	all	A
P11	all	B

Figure 6: Summary of the training and testing environments used in the experiments which are represented using the interventions performed on the environment.

other possible explanation is that it may simply not be possible to find a strategy which simultaneously works for all parameters (note that the agent does not have access to the randomized parameters and hence must be robust to them). This poses an interesting question for future work.

Given the above experiments, we can see that an agent's generalization capabilities are related to the experience gathered under its training curriculum. Our evaluation protocols allow us to explore this relationship in a differentiated manner, assessing which curricula lead to which generalization abilities. This will not only help uncover an agent's shortcomings but will likewise aid in investigating novel approaches for robustness in RL. Lastly, we want to note that this benchmark comprises extremely challenging tasks that appear to be out of reach for model free methods without any additional inductive bias.

5 Conclusion

We have introduced a new benchmark - `CausalWorld` - to accelerate research in causal structure and transfer learning using a simulated environment of an open source robot, where skills learned could potentially be transferred to the real world. We showed how enabling do-interventions on the environment's properties yields a natural way of defining learning curricula and evaluation protocols that can disentangle different generalization capabilities. We showed how different training curricula can affect generalization capabilities with respect to different environment properties. Finally, we hope that the modular and flexible formulation of the environments in `CausalWorld` will benefit research into a wide range of topics related to transfer in reinforcement learning and causal structure learning.

References

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528*, 2019.

Victor Bapst, Alvaro Sanchez-Gonzalez, Carl Doersch, Kimberly L Stachenfeld, Pushmeet Kohli, Peter W Battaglia, and Jessica B Hamrick. Structured agents for physical construction. *arXiv preprint arXiv:1904.03177*, 2019.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Yvonne M Caldera, Anne McDonald Culp, Marion O'Brien, Rosemarie T Truglio, Mildred Alvarez, and Aletha C Huston. Children's play preferences, construction play with blocks, and visual-spatial skills: Are they related? *International Journal of Behavioral Development*, 23(4):855–872, 1999.

Beth M Casey, Nicole Andrews, Holly Schindler, Joanne E Kersh, Alexandra Samper, and Juanita Copley. The development of spatial skills through interventions involving block building activities. *Cognition and Instruction*, 26(3):269–309, 2008.

Stephanie CY Chan, Sam Fishman, John Canny, Anoop Korattikara, and Sergio Guadarrama. Measuring the reliability of reinforcement learning algorithms. *arXiv preprint arXiv:1912.05663*, 2019.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: A platform to study the sample efficiency of grounded language learning. In *International Conference on Learning Representations*, 2018.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

Erwin Coumans et al. Bullet real-time physics simulation. *URL http://bulletphysics. org*, 2013.

Gwen Dewar. The benefits of toy blocks: The science of construction play. *Parentig Science*, 2018.

Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. An empirical investigation of the challenges of real-world reinforcement learning. *arXiv preprint arXiv:2003.11881*, 2020.

Scott Elliott Fahlman. A planning system for robot construction tasks. *Artificial intelligence*, 5(1): 1–49, 1974.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018.

Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck. *arXiv preprint arXiv:1901.10902*, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

Peter Henderson, Wei-Di Chang, Florian Shkurti, Johanna Hansen, David Meger, and Gregory Dudek. Benchmark environments for multitask learning in continuous domains. *arXiv preprint arXiv:1708.04352*, 2017a.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *arXiv preprint arXiv:1709.06560*, 2017b.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

Michael Janner, Sergey Levine, William T Freeman, Joshua B Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. *arXiv preprint arXiv:1812.10972*, 2018.

Constance Kamii, Yoko Miyakawa, and Yasuhiko Kato. The development of logico-mathematical knowledge in a block-building activity at ages 1–4. *Journal of Research in Childhood Education*, 19(1):44–57, 2004.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Youngwoon Lee, Edward S. Hu, Zhengyu Yang, Alex Yin, and Joseph J. Lim. Ikea furniture assembly environment for long-horizon complex manipulation tasks, 2019.

Will McCarthy, David Kirsh, and Judith Fan. Learning to build physical structures better over time.

Mehdi Mirza, Andrew Jaegle, Jonathan J. Hunt, Arthur Guez, Saran Tunyasuvunakool, Alistair Muldal, Théophane Weber, Peter Karkus, Sébastien Racanière, Lars Buesing, Timothy Lillicrap, and Nicolas Heess. Physically embedded planning problems: New challenges for reinforcement learning, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Swiya Nath and Dénes Szücs. Construction play and cognitive skills associated with the development of mathematical abilities in 7-year-old children. *Learning and Instruction*, 32:73–80, 2014.

Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepezvari, Satinder Singh, et al. Behaviour suite for reinforcement learning. *arXiv preprint arXiv:1908.03568*, 2019.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *arXiv preprint arXiv:1703.02702*, 2017.

Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. Towards generalization and simplicity in continuous control. In *Advances in Neural Information Processing Systems*, pp. 6550–6561, 2017.

Miles Richardson, Thomas E Hunt, and Cassandra Richardson. Children's construction task performance and spatial ability: Controlling task complexity and predicting mathematics performance. *Perceptual and motor skills*, 119(3):741–757, 2014.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Brian N Verdine, Roberta Michnick Golinkoff, Kathy Hirsh-Pasek, and Nora Newcombe. *Links between spatial and mathematical skills across the preschool years*. Wiley, 2017.

Shimon Whiteson, Brian Tanner, Matthew E Taylor, and Peter Stone. Protecting against evaluation overfitting in empirical reinforcement learning. In *2011 IEEE symposium on adaptive dynamic programming and reinforcement learning (ADPRL)*, pp. 120–127. IEEE, 2011.

Terry Winograd. Understanding natural language. *Cognitive psychology*, 3(1):1–191, 1972.

Patrick H Winston. Learning structural descriptions from examples. 1970.

Manuel Wüthrich, Felix Widmaier, Felix Grimminger, Joel Akpo, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, et al. Trifinger: An open-source robot for learning dexterity. *arXiv preprint arXiv:2008.03596*, 2020.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. *arXiv preprint arXiv:1910.10897*, 2019.

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

6 APPENDIX

A OBSERVATIONS

Observations in `CausalWorld` has two modes, "structured" and "pixel". When using "pixel" mode, 6 images are returned consisting of the current images rendered from 3 different views on top of the TriFinger platform, showing the current state of the environment, as well as the 3 equivalent goal images rendered from the same points of view, showing the goal shape that the robot have to build by the end of the episode.

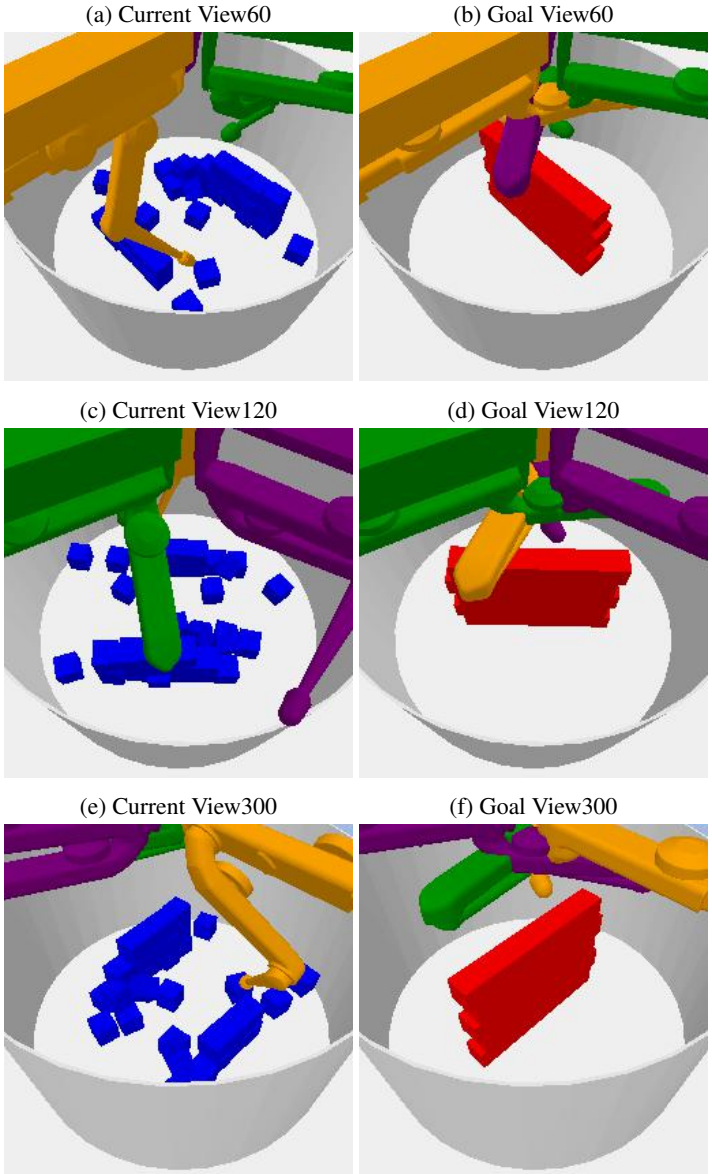

Figure 7: Example "pixel" mode observations returned at each step of the environment.

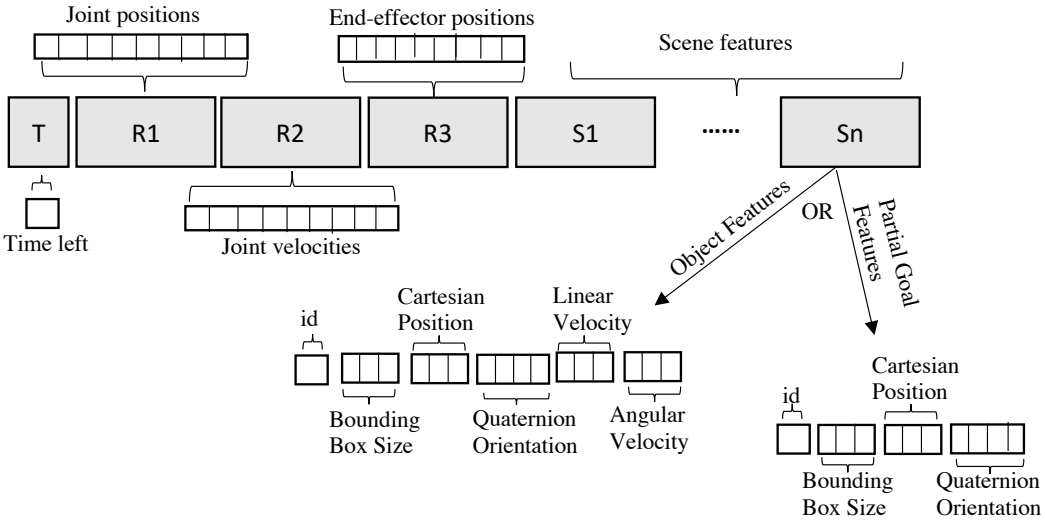

Figure 8: Structured observation description. For the scene features, all the blocks feature vector are concatenated first. Following that the partial goals feature vector are concatenated in the same order. Lastly, if there is any obstacles/ fixed blocks, their feature vectors are concatenated at the end following the same description as the partial goal features.

B TRIFINGER PLATFORM

The robot from (Wüthrich et al., 2020) shown in figure 9 is open-sourced and can be reproduced and built in any research lab; since its inexpensive (about $5000), speeding up sim2real research.

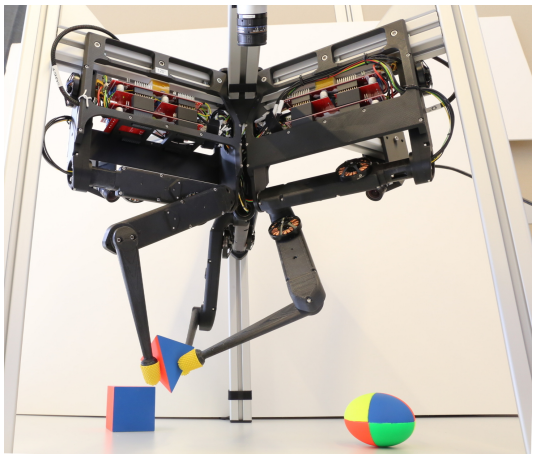

Figure 9: The TriFinger platform.

C TASK GENERATORS

1. *Pushing*: task where the goal is to push one block towards a goal position with a specific orientation; restricted to goals on the floor level.
2. *Picking*: task where the goal is to pick one block towards a goal height above the center of the arena; restricted to goals above the floor level.

3. *Pick And Place*: task where the arena is divided by a fixed long block and the goal is to pick one block from one side of the arena to a goal position with a variable orientation on the other side of the fixed block.
4. *Stacking2*: task where the goal is to stack two blocks above each other in a specific goal position and orientation.
5. *Towers*: task where the goal is to stack multiple n blocks above each other in a specific goal position and orientation - exactly above each other creating a tower of blocks.
6. *Stacked Blocks*: task where the goal is to stack multiple n blocks above each other in an arbitrary way to create a stable structure. The blocks don't have to be exactly above each other; making it more challenging than the ordinary towers task since the its harder to come up with a stable structure that covers the goal shape volume.
7. *Creative Stacked Blocks*: exactly the same as the *Stacked Blocks* task except that the first and last levels of the goal are the only levels shown or "imposed" and the rest of the structure is not explicitly specified, leaving the rest of the goal shape to the imagination of the agent itself; this is considered the most challenging since its it needs the agent to understand how to build stable structures and imagine what can be filled in the middle to connect the two levels in a stable way.
8. *General*: the goal shape is an arbitrary shape created by initially dropping an arbitrary number of blocks from above the ground and waiting till all blocks come to a rest position where this becomes the goal shape that the agent needs to fill up afterwards.

Variable	Sub Variable	Space $\mathbf{A}$	Space $\mathbf{B}$
gravity[z]	-	$[-10, -7]$	$[-7, -4]$
floor friction	-	$[0.3, 0.6]$	$[0.6, 0.8]$
stage friction	-	$[0.3, 0.6]$	$[0.6, 0.8]$
stage color [rgb]	-	$[0.0, 0.5]^3$	$[0.5, 1]^3$
floor color [rgb]	-	$[0.0, 0.5]^3$	$[0.5, 1]^3$
joint positions	-	$[[-1.57, -1.2, -3.0]^3,$ $[-0.69, 0, 0]^3]$	$[[-0.69, 0, 0]^3,$ $[1.0, 1.57, 3.0]^3]$
block	size	$[0.055, 0.075]^3$	$[0.075, 0.095]^3$
block	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
block	mass	$[0.015, 0.045]$	$[0.045, 0.1]$
block	position (cylindrical)	$[[0, -\pi, h/2],$ $[0.11, \pi, 0.15]]$	$[[0.11, -\pi, h/2],$ $[0.15, \pi, 0.3]]$
goal cuboid	size	$[0.055, 0.075]^3$	$[0.075, 0.095]^3$
goal cuboid	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
link	color	$[0.0, 0.5]^3$	$[0.5, 1]^3$
link	mass	$[0.015, 0.045]$	$[0.045, 0.1]$

Table 2: Description of a subset of the high level variables, exposed in CausalWorld, and their corresponding spaces, h refers to the height of the block.

Task Generator	Variable	Space $\mathbf{A}$	Space $\mathbf{B}$
Picking	goal height	$[0.08, 0.20]$	$[0.20, 0.25]$
Towers	tower dims	$[[0.08, 0.08, 0.08],$ $[0.12, 0.12, 0.12]]$	$[[0.12, 0.12, 0.12],$ $[0.20, 0.20, 0.20]]$

Table 3: Example of task generators' specific high level variables, exposed in CausalWorld, and their corresponding spaces. For a full list of each task generators' variables and their corresponding spaces, please refer to the documentation at (https://sites.google.com/view/causal-world/home).

Task generators	Dense reward
Pushing	$-750\Delta^t(o_1, e) - 250\Delta^t(o_1, g_1)$
Picking	$-750\Delta^t(o_1, e) - 250\Delta^t(o_{1,z}, g_{1,z}) - 125\Delta^t(o_{1,x,y}, g_{1,x,y}) - 0.005\|\|v^t - v^{t-1}\|\|$
Pick and Place	$-750\Delta^t(o_1, e) - 50\Delta^t(o_{1,x,y}, g_{1,x,y}) - 250(\|o^t_{1,z} - t\| - \|o^{t-1}_{1,z} - t\|) - 0.005\|\|v^t - v^{t-1}\|\|$
Stacking	$\mathbf{1}_{d^t(o_1,e)>0.02}(-750\Delta^t(o_1, e) - 250\Delta^t(o_1, g_1)) + \mathbf{1}_{d^t(o_1,e)<0.02}(-750\Delta^t(o_2, e) - 250(\|o^t_{2,z} - g^t_{2,z}\| - \|o^{t-1}_{1,z} - g^t_{2,z}\|) - \mathbf{1}_{o^t_{2,z} - g^t_{2,z} > 0} 125\Delta^t(o_{2,x,y}, g_{2,x,y})) - 0.005\|\|v^t - v^{t-1}\|\|$

Table 4: Description of the dense rewards applied in our experiments. The following notation was applied: $v^t \in \mathbf{R}^3$ joint velocities, $e^t_i \in \mathbf{R}^3$ i-th end-effector positions, $o^t_i \in \mathbf{R}^3$ i-th block position, $g^t_i \in \mathbf{R}^3$ i-th goal block position, $d^t(o, e) = \sum_i ||e^t_i - o^t||$ the distance between end-effectors and the block, $\Delta^t_{o,e} = d^t(o, e) - d^{t-1}(o, e)$ the distance difference w.r.t. the previous timestep. The target height parameter t for pick and place is 0.15 if block and goal are of different height. Otherwise, t is half the goal height.

D TRAINING DETAILS

The experiments were carried out using the `stable baselines` implementation of PPO, SAC and TD3. We used a 2 layer MLP Policy [256,256] for all the policies. PPO was trained on 20 workers up to 100 million timesteps in parallel and SAC as well as TD3 were trained serially for 10 million timesteps.

PPO		SAC		TD3	
discount	0.99	discount	0.95	discount	0.96
batch size	120000	entropy coeff	1e-3	batch size	128
learning rate	2.5e-4	batch size	256	learning rate	1e-4
entropy coef.	0.01	learning rate	1e-4	buffer size	500000
value function coef.	0.5	target entropy	auto	tau	0.02
gradient clipping (max)	0.5	buffer size	1000000		
n minibatches per update	40	tau	0.001		
n training epochs	4				

Table 5: Learning algorithms hyper parameters used in the baselines experiments.

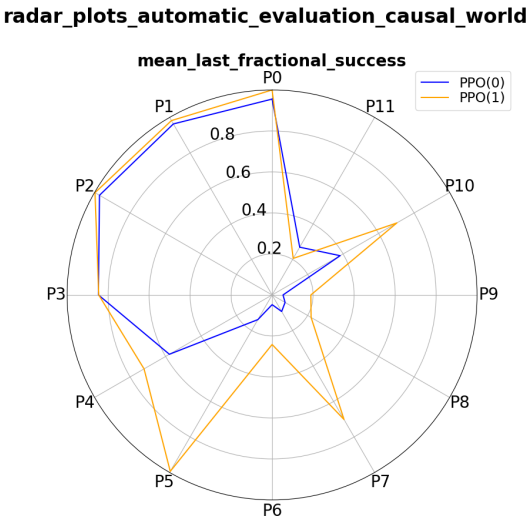

Figure 10: An example of model selection in `CausalWorld` by evaluating generalization across the various axes using the previously mentioned protocols. Here we compare two agents trained on different curricula using PPO.

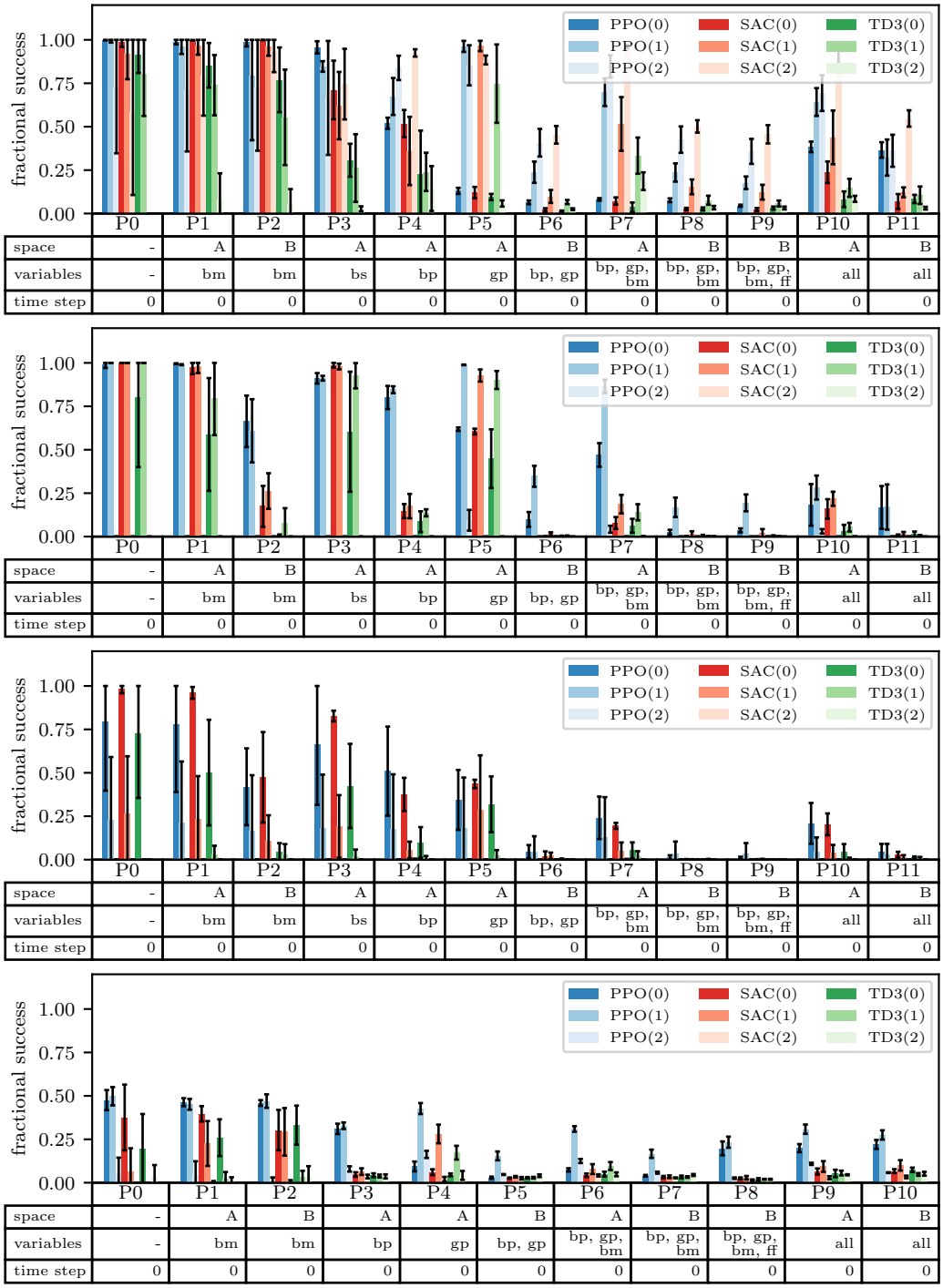

Figure 11: Evaluation scores, for pushing, picking, pick and place and stacking2 baselines, from top to bottom respectively. Each protocol was evaluated for 200 episodes and each bar is averaged over five models with different random seeds [bp block pose, bm block mass, bs block size, gp goal pose, ff floor friction].