LEARNING TO INFER UNSEEN CONTEXTS IN CAUSAL CONTEXTUAL REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

In Contextual Reinforcement Learning (CRL), a change in the context variable can cause a change in the distribution of the states. Hence contextual agents must be able to learn adaptive policies that can change when a context changes. Furthermore, in certain scenarios agents have to deal with unseen contexts, and be able to choose suitable actions. In order to generalise onto unseen contexts, agents need to not only detect and adapt to previously observed contexts, but also reason about how a context is constructed, and what are the causal factors of context variables. In this paper, we propose a new task and environment for Causal Contextual Reinforcement Learning (CCRL), where the performance of different agents can be compared in a causal reasoning task. Furthermore, we introduce a *Contextual Attention Module* that allows the agent to test different interventions, and discover the causal factors. Finally, we demonstrate that non-causal agents fail to generalise onto unseen contexts, while the agents incorporating the proposed module can achieve better performance in unseen contexts.

1 INTRODUCTION

In Reinforcement Learning, an agent interacts with an environment through observations and actions, and tries to maximise the cumulative reward that is defined for a given task. However, certain scenarios can cause the distribution of the observations and reward to change over time, which often results in performance degradation in agents that fail to adapt to these changes.

In Contextual Reinforcement Learning (CRL), a change of context affects the distribution of the environments' observations and the reward distribution. As such changes may occur numerous times, not only does the agent have to *identify* the change of context, but it also has to adapt to it, while *remembering* the previous contexts. This problem is known as Contextual Reinforcement Learning (CRL).

In this settings, inferring contexts is of high importance, as it assists the agent in adapting its policy accordingly. However, the CRL settings that have been studied in the literature often use fixed contexts, and the agent only has to detect, and adapt to contexts that were observed during training. This scenario is not very realistic, as in the real world, contexts are ever-changing, and not only does an agent have to learn the *training* contexts, but it also has to generalize to *unseen contexts*. For this purpose, the agent has to *understand* the concept of a context, and *reason* on how a context can be inferred.

For example, consider a setting where an agent is a websites' recommender system, and the environment is the user. In a scenario where the user is looking for a specific product, for example a laptop bag, if the agent suggests to the user a laptop, the user would consider this an error in the recommender system (because the user presumably already has a laptop, which is why she is looking for a bag). To avoid this, the agent has to first realize the relationship between the bag and the laptop, and determine that the laptop is the cause in this interaction, and the bag is an effect, and not the other way around (Schölkopf et al., 2021). This is a challenging problem that requires *reasoning capabilities*, and discovering causal factors, in order to generalise onto unseen contexts. For example, assuming that the agent was trained with the laptop bag interaction, and later a user looked for a guitar bag, the agent should not recommend the user a guitar. Because of the importance of causal factors, we will refer to this setting as *Causal Contextual Reinforcement Learning (CCRL)* in the remainder of this paper.

In this paper, we investigate how RL agents can learn to infer, and generalise to unseen contexts. Agents that only *memorise* the contexts may be successful on the contexts observed during training, but will fail to generalize when dealing with unseen contexts. To study this, we provide a simple environment that targets evaluation of reasoning capabilities in agents for inferring context variables, and compare generalisation of different agents on both *seen* and *unseen* contexts. We further demonstrate that using disentangled representations, can help an agent discover the causal factors of the context variable, enabling it to better generalise onto unseen contexts.

In summary, our contributions are as follows:

1. We propose a new CCRL task that can only be solved if agents are capable of reasoning about the construction of contexts.
2. We provide a simple environment designed to address this problem, and evaluation measures for comparing the performance of agents on this task.
3. We demonstrate that in our environment, State-Of-The-Art (SOTA) agents struggle to generalise onto unseen contexts, as they are unable to discover the causal factors of the context variable.
4. Finally, we propose a *Contextual Attention Module* that allows an agent to test different interventions and helps discovering the causal factors, exceeding the performance of SOTA on both seen, and unseen contexts.

2 Related Work

Several approaches address CRL under the fixed context setting. For example, models that find better exploration strategies (Gregor et al., 2016; Pathak et al., 2019), have been used to tackle environments with changing dynamics. As the context changes, exploiting the current policy is no longer as effective and thus not suitable to tackle the changes in the environment. Hence, the agent needs to explore new actions, in order to accumulate more reward. Another approach to adapt to new contexts, is to use options in a hierarchical RL setting, where a meta-policy switches between a set of available policies (Achiam et al., 2018; Eysenbach et al., 2018). Hallak et al. (2015) define a Contextual Markov Decision Process (CMDP) as a constrained Partially Observable Markov Decision Process (POMDP), where each context is parameterised as an MDP. They propose a solution to tackle CRL assuming a fixed observation space over different contexts, where the agent needs to pick a suitable policy, given the available context. Jiang et al. (2017) propose a generalisation of MDPs and POMDPs known as Contextual Decision Processes (CDPs), where there is a general context space that the observations are drawn from. Although this formulation is quite general, their work focuses on problems with low Bellman ranks, which corresponds to MDPs with low-rank transition matrix, or small observation space.

In non-contextual RL settings, several works have been devoted to studying causality. Ha & Schmidhuber (2018) introduce generative recurrent world models to capture some of the causal relations underlying the environment. Buesing et al. (2018) show that incorporating counterfactual reasoning can improve the data efficiency of RL algorithms. Goyal et al. (2019) propose a new recurrent architecture in which multiple recurrent cells communicate through an attention layer, and compete with each other such that they only update when they are most relevant. They report that such architectures improve generalisation in partially observable environments, and sparse reward conditions. Additionally, some environments have been proposed to investigate causality in RL. Ahmed et al. (2020) introduce a robotics environment to investigate how well an agent will perform on different evaluation distributions, depending on the curriculum it has been trained with. Higgins et al. (2017) propose an environment that studies the generalisation of agents to new unseen variables.

In our work, we focus on the CCRL setting, and propose a simple RL environment targeted at contextual settings and designed to evaluate causal decision making performance of agents. Our environment is also computationally efficient, hence accessible to a larger audience. We investigate approaches that target disentangled representations in RL (Higgins et al., 2017), and show how effective such solutions are on our reasoning CCRL task. Furthermore, we limit ourselves to a simpler non-recurrent setup, where models require reasoning to solve the task, without sequential information.

3 PROBLEM DEFINITION

As discussed in the example given in Section 1, in CCRL an agent has to learn a causal model, and discover the underlying factors of why a user is looking for a laptop bag. This kind of generalisation is different from representational generalisation, where the agents' state representation has to generalise to unseen states (e.g, the agent has to recognize a new user from her queries). In this latter case, the realisation of a dependency between factors can be modelled merely using statistical models with an approximation on the conditional probability of variables. However in the former case (laptop bag), discovering the causal factors requires a causal model.

To learn a causal model that is useful for such scenarios, the agent has to determine which interventions are allowed, and which are useful to the task at hand (Schölkopf et al., 2021). To this end, the agent needs to not only discover the disentangled factors of variation (e.g, shapes, sizes, colours), but also *test* different interventions of these variables, to discover the causal factors.

In Section 4, we propose a simple environment that can only be solved if an agent learned the underlying causal factors of the task. We will show that our task is difficult to solve for non-causal SOTA agents, and only agents that reason about the context variables can be successful in solving it. We will also show that exploration and data efficiency alone are not enough to succeed in this scenario, and causal reasoning is required as an orthogonal aspect. Furthermore, in Section 5 we propose *Contextual Attention Module*, a method that enables agents to test interventions, resulting in a model that is more capable of generalising onto unseen contexts in a CCRL setting. We will show that statistical models, even those based on disentangled factors, struggle to generalise to unseen contexts, as they do not have a mechanism to test interventions; while the proposed approach is more successful.

4 THE PROPOSED ENVIRONMENT

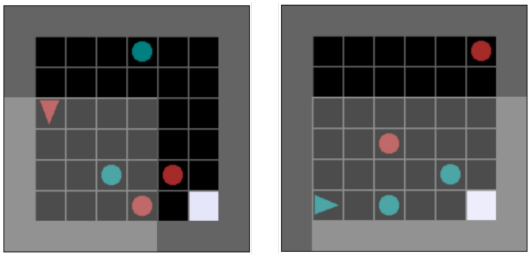

Figure 1: Contextual reasoning grid-world example of a context pair. While colours are shared across both contexts, their meaning differs based on the agent colour. The agent (triangle) only sees a small excerpt of the full environment depending on its position and orientation (indicated as the highlighted region in the figure). The white square is the designated goal position.

To investigate the CCRL setup as described in Section 3, we create a grid-world environment based on the work by Chevalier-Boisvert et al. (2018). In this environment, agents are trained on a set of *training* contexts, and are evaluated on two different sets of *testing contexts*. The aim is to evaluate how well agents captured the causal variables from the training, and can reason to correctly infer and generalise to new (test) contexts. In other words, learning training contexts successfully, without discovering the causal factors, is not useful for succeeding in testing contexts.

The task of the agent is to reach a random goal position in an 8×8 grid. Furthermore, the environment contains two *goodie* objects and two *obstacle* objects, which the agent has to collect, and avoid, respectively. Collecting a goodie yields a reward of $+1$, whereas collecting an obstacle results in a -1 reward. For reaching the goal position, the agent receives a reward of 1 that is linearly discounted by the number of time steps it took to reach it. If the agent is not able to reach the goal within 100 steps, an episode ends and the agent receives a reward of 0. Observations are given as 84×84 RGB images representing excerpts of the environment depending on the current orienta-

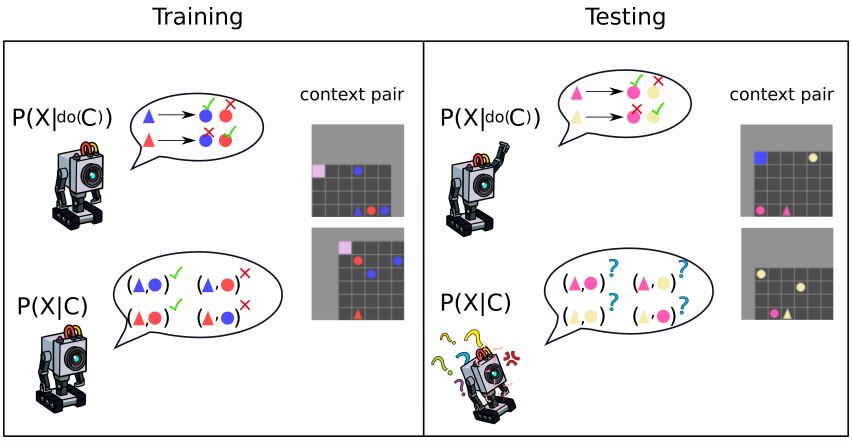

Figure 2: A visual explanation of the proposed CCRL task, and a statistical agent denoted by $P(X \mid C)$ vs a causal agent denoted by $P(X \mid \text{do}(C))$[1].

tion of the agent. The observation encodes the goal as a rectangle, goodies/obstacles as circles with different colours, and the agent as a triangle (see Figure 1).

To incorporate the contextual aspect, we rely on different colour combinations for goodies, obstacles, goal and agent. While the goal colours are arbitrarily chosen, we set the goodie colour to be the same as the agent, and obstacle has always a different colour than the goodie. This allows the agent to infer what it should collect or avoid, respectively. Since we want to study the reasoning capabilities of our agents, we cannot simply define distinct contexts with non-overlapping colours, as the agent could remember which colours were good or bad. Thus, we create *context pairs* that share the same colours, but swap the obstacle and goodie/agent colours. Contexts are uniformly sampled at the beginning of each episode. In Figure 2, we provide an example to illustrate our task, and the behaviour of a non-causal vs. a causal agent. On the left, we see that both agents deal with various context pairs, and learn to solve the task under training contexts. While the causal agent discovers the causal factors of the context (e.g, the colour of the agent determines the colour of the goodie), the non-causal agent only *memorises* what combinations of agent and goodie/obstacle colours are good or bad. Although they function in different ways, both succeed in solving the task under training contexts. Under the test contexts however (as shown in Figure 2, right) only a causal agent can solve the task. For example, one pair could be defined as *goal*: pink, *obstacle*: blue, *goodie/agent*: red, and *goal*: pink, *obstacle*: red and *goodie/agent*: blue. Thus, the agent cannot just learn to collect red circles since those are only considered to be goodies in the first context, but it has to learn *the relation between its own and the goodie colour*.

In our experiments, the agents will be trained on a set of 4 context pairs (8 contexts), and evaluated on two different sets of test context pairs comprising unseen obstacle and agent/goodie colour combinations. In the first set, we reuse the colours from the training set, but create different obstacle and agent/goodie combinations, which have not been seen before. All in all 24 new context pairs (48 contexts) will be used in this *seen-colour* scenario. For the second set, we use new colours for obstacles, agent and goodies that were not used during training to create 28 new context pairs (56 contexts). We will refer to this as the *unseen-colour* scenario. These setup ensures the contextual aspect of CCRL, while only allowing a causal agent to succeed in the test environment. Hence, it is a perfect test bed for CCRL. The details regarding the context configurations used for the experiments in this paper are provided in the Appendix A.3; note that our environment and the tasks can be easily modified and extended by changing the context configurations.

[1] $P(X \mid \text{do}(C))$ refers to a model that considers different interventions based on the context variable; hence a do-calculus notation is used (Pearl, 2012).

5 CONTEXTUAL ATTENTION MODULE

In order to discover the causal factors affecting the context, we propose the *Contextual Attention Module (CAM)*, capable of incorporating contextual information into an attention layer as keys, allowing the model to attend on relevant features. Furthermore, this module enables the model to test various interventions based on the context variables, presented by a set of disentangled factors (e.g, shape,size, colour) , and discover the causal factors. To this end, we incorporate disentangled representations, that represent causal mechanisms (Schölkopf, 2019); enabling the model to test different interventions between these causal factors, and the state features, using an attention mechanism.

Let $\boldsymbol{x} \in \mathbb{R}^{C \times N}$ be a set of state feature maps from the previous convolutional layer, and $\boldsymbol{c} \in \mathbb{R}^{1 \times D}$ be a vector of disentangled factors, representing the context variable. We transform $\boldsymbol{x}$ and $\boldsymbol{c}$ into feature spaces $\boldsymbol{f}, \boldsymbol{g}$ as query and key, respectively, and further calculate the attention maps, with:

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W_f}\boldsymbol{x} + b_f, \ \boldsymbol{g}(\boldsymbol{c}) = \boldsymbol{W_g}\boldsymbol{c} + \boldsymbol{b_g}$$
$$\beta_{j,i} = \frac{\exp(s_{ij})}{\sum_{i=1}^{D} \exp(s_{ij})}, \text{ where } s_{ij} = \boldsymbol{g}(\boldsymbol{c_i})\boldsymbol{f}(\boldsymbol{x_j})^T, \tag{1}$$

and $\beta_{j,i}$ indicates the extent to which the model attends to the jth visual feature, by the ith factor of variation. The output of the attention layer is $\boldsymbol{o} = (\boldsymbol{o_1}, \boldsymbol{o_2}, ..., \boldsymbol{o_j}, ..., \boldsymbol{o_N}) \in \mathbb{R}^{C \times N}$, where,

$$\boldsymbol{o_j} = \sum_{i=1}^{N} \beta_{j,i}\boldsymbol{h}(\boldsymbol{x_i}), \ \boldsymbol{h}(\boldsymbol{x_i}) = \boldsymbol{W_h}\boldsymbol{x_i} + \boldsymbol{b_h}. \tag{2}$$

and $\boldsymbol{W_g} \in \mathbb{R}^{D' \times D}$, $\boldsymbol{W_f} \in \mathbb{R}^{1 \times C}$, and $\boldsymbol{W_h} \in \mathbb{R}^{C \times C}$ are weight matrices; D' is the dimensionality of the intermediate features that matches $f(x)$ dimensionality (in our case, 25); and $\boldsymbol{b_g}$, $\boldsymbol{b_f}$, and $\boldsymbol{b_h}$ are the bias terms. We also multiply the output of the attention layer by a scale parameter and add back the input feature map:

$$\boldsymbol{y_i} = \gamma \boldsymbol{o_i} + \boldsymbol{x_i}, \tag{3}$$

where γ is a learnable scalar initialized by 0. The parameter γ allows the network to first learn useful features from the states, and then focus on attending to the disentangled factors by assigning more weights to the contextual evidence; and is reported to improve learning speed (Zhang et al., 2019).

To compute disentangled factors, CAM can incorporate a feature disentanglement technique. In the experiments reported in this paper, we use a pre-trained annealed VAE (Burgess et al., 2018) to compute $\boldsymbol{c}$ given the state.

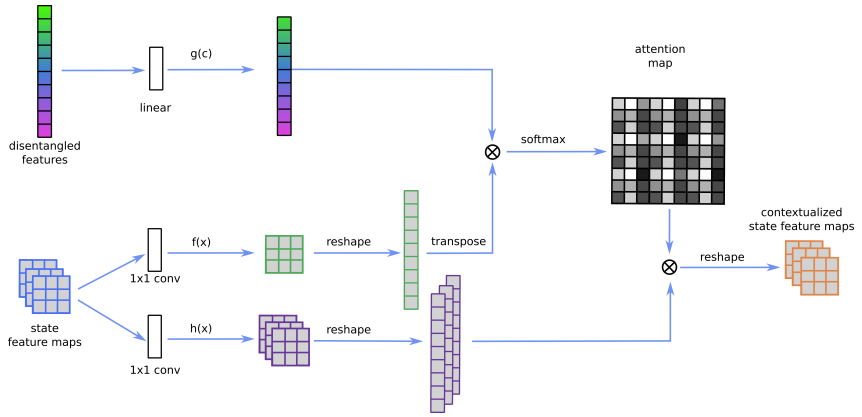

Figure 3: The Contextual Attention Module (CAM).

6 EXPERIMENTAL DESIGN

In this section, we introduce our training setup and detail the proposed evaluation measures for our proposed environments.

6.1 TRAINING

We train 5 PPO agents (Schulman et al., 2017) with different backend setups (i.e., different state and context encoder combinations): *Non-Contextual* (*NC*), *Self Attention* (*SA*), *Darla* (Higgins et al., 2017), *Disentangled Concatenated Features* (*DACat*), and *Contextual Attention Module* (*CAM*, see Section 5). Experiments are repeated with 8 different random seeds.

For *NC*, we use a simple state encoder consisting of 4 convolutional layers followed by a fully-connected layer (see Appendix A.1 for a complete architecture specification and further training details). For *SA*, we use the same 4 layer CNN to encode the state, however the fully-connected layer is replaced by a self-attention mechanism (Zhang et al., 2019). For *Darla*, we use the encoding network of a pre-trained annealed VAE (Burgess et al., 2018) as the state encoder, which should provide the policy with a disentangled state representation. The weights of the encoder are not updated during training. Training details as well as some reconstructed samples from this VAE are provided in Appendix A.2.

While the aforementioned backends do not utilize separate networks to infer contexts, *DACat* and *CAM* rely on disentangled representations to incorporate contextual information. Both of these methods use the same pre-trained VAE encoder as *Darla* with frozen weights to create disentangled representations of the observations which will be used as context variables. To combine visual and contextual information, *DACat* performs a simple concatenation of the disentangled features and the visual features that are encoded using the same architecture as *NC*. In contrast, *CAM* relies on an attention mechanism that uses the disentangled representation to attend to relevant visual features of the state (see Section 5). Visual features are created using the same 4 layer CNN as *NC* and *SA*.

6.2 EVALUATION

For evaluation, we consider three metrics: goal reached (G), remaining goodies (RG), and remaining obstacles (RO), where RG and RO are normalized to $[0, 1]$ by dividing with the number of goodies and obstacles, respectively. These are computed for each context separately and averaged across 50 episodes. For G and RO, higher is better, and for RG lower is better.

Additionally, we combine these metrics to a single score value for each context c:

$$S^c = w_G \cdot G^c + w_{OG} \cdot \max\left(0, RO^c - RG^c\right), \tag{4}$$

where w_G, w_{OG} are weights that control the importance of reaching goal, as well as the correct interaction with goodies and obstacles. The maximum operator ensures that only agents executing the correct interaction with goodie/obstacle increase score, and *collector*[2], *ignorer*[3], and *illogical*[4] agents are equally punished in evaluation.

Using this score, we now compute a *Solved Context-Pair Ratio (SCPR)* for the set of all context pairs $(c_i, c_j) \in C$, by considering only if both contexts in a pair were solved, based on a given threshold value t:

$$\text{SCPR} = \frac{1}{|C|} \sum_{(c_i, c_j) \in C} \mathbb{1}_{\min\left(S^{c_i}, S^{c_j}\right) \geq t}, \tag{5}$$

with $\mathbb{1}_{\min\left(S^{c_i}, S^{c_j}\right) \geq t}$ yielding 1 if $\min\left(S^{c_i}, S^{c_j}\right) \geq t$ and 0 otherwise. Finally, we create three different evaluation scores by changing the weights w_G, w_{OG} to assess and compare the behaviour of our agents:

First, **Goodie/Obstacle Discrimination per Context-Pair (GOD-PCP)** for evaluating the ability of agents in distinguishing between obstacles and goodies in each context pair, and defined as: GOD-PCP $= SCPR, w_{OG} = 1, w_G = 0$.

Second, **Reached-Goal per Context-Pair (RG-PCP)** for measuring the agents' performance in arriving at the goal; defined by: RG-PCP $= SCPR, w_{OG} = 0, w_G = 1$.

Third, **Average SCPR** by considering both obstacle/goodie mixup and arrival at the goal, and is calculated by: Average SCPR $= SCPR, w_{OG} = 0.5, w_G = 0.5$.

[2] A *collector* agent is an agent that collects both goodie and obstacles.

[3] An *ignorer* agent is an agent that ignores both goodie and obstacles.

[4] An *illogical* agent is an agent that ignores goodie and collects obstacles.

7 RESULTS

In Figure 4, we plot the three evaluation scores as introduced in Section 6, for different thresholds, both for contexts in *training*, *test with seen*, and *test with unseen colours*. Looking at GOD-PCP, we observe that the non-contextual (*NC*) as well as the *CAM* agent are able to solve the training context pairs, even for high threshold values. Also RG-PCP shows that both *NC* and *CAM* agents arrive almost always at the goal, while other agents fail to do so. Furthermore, the Average-SCPR score represents the overall success of *NC* and *CAM* agents in solving the training contexts (Figure 4 first row).

However, on the test contexts the performance of all agents in distinguishing obstacle from goodie substantially degrades as reported by GOD-PCP. Though overall, agents perform similarly in arriving at the goal in the test contexts. While *NC* and *CAM* remain best performing agents on test contexts, we observe that *CAM* performs better than *NS* in all measures, on both seen, and unseen-colour scenarios. These results suggest a better generalisation in terms of causal reasoning (measured by GOD-PCP), and overall performance (as measured by RG-PCP and Average-SCPR) for the *CAM* agent (Figure 4 second and third rows). Also the poor performance of *Darla* and *DACat* agents suggests that only providing disentangled representations to the policy is not sufficient to achieve better causal generalisation. Likewise, the results of *SA* indicate that incorporating attention mechanism alone can not improve the results either, further suggesting *CAM* as a better solution for causal reasoning.

Table 1 summarizes the results for the three metrics – *goal reached*, *remaining goodies* and *remaining obstacles* – separately for the train and test context sets. Results are averaged across all context pairs and random seeds. We observe that all five methods are able to distinguish between goodies and obstacles to some extent on the train set, with the *NC* and *CAM* agents performing best. Similar to Figure 4, the performance degrades for unseen contexts. While there is a high variance across all agents, the proposed *CAM* model achieves the best average scores as well as the least spread for *goal reached* and *remaining goodies* in each context setting.

All in all, based on the provided evaluations, we observe that our proposed environment is a challenging task for SOTA RL agents, and there is a considerable generalisation gap to be filled by incorporating causal reasoning into agents.

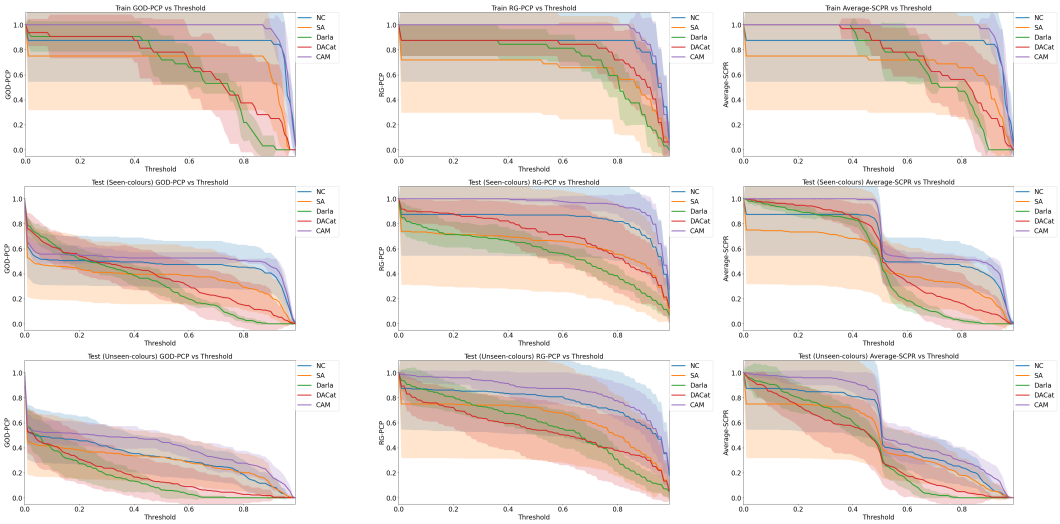

Figure 4: Evaluation scores for: train (first row), test with seen-colours (second row), and test with unseen-colours (third row).

Table 1: Average performance across all context pairs and seeds for *Remaining Goodie*, *Remaining Obstacle* and *Goal Reached*. We report mean and standard deviation on train, test with seen-colours, and test with unseen-colours. Best results based on the mean score are marked bold. For *Remaining Goodie* lower and for *Remaining Obstacle* and *Goal Reached* higher is better.

		Remaining Goodie	Remaining Obstacle	Goal Reached
train	NC	0.158 ± 0.324	$\mathbf{0.994 \pm 0.008}$	0.832 ± 0.321
	SA	0.300 ± 0.411	0.990 ± 0.016	0.632 ± 0.415
	Darla	0.292 ± 0.200	0.855 ± 0.092	0.707 ± 0.300
	DACat	0.268 ± 0.218	0.909 ± 0.081	0.777 ± 0.311
	CAM	$\mathbf{0.042 \pm 0.026}$	$\mathbf{0.994 \pm 0.009}$	$\mathbf{0.944 \pm 0.038}$
seen	NC	0.173 ± 0.314	0.615 ± 0.449	0.818 ± 0.320
	SA	0.318 ± 0.403	$\mathbf{0.654 \pm 0.420}$	0.625 ± 0.413
	Darla	0.344 ± 0.151	0.557 ± 0.307	0.554 ± 0.371
	DACat	0.295 ± 0.211	0.585 ± 0.315	0.699 ± 0.324
	CAM	$\mathbf{0.058 \pm 0.035}$	0.546 ± 0.464	$\mathbf{0.944 \pm 0.089}$
unseen	NC	0.346 ± 0.307	0.550 ± 0.410	0.760 ± 0.339
	SA	0.418 ± 0.373	$\mathbf{0.634 \pm 0.404}$	0.621 ± 0.386
	Darla	0.537 ± 0.204	0.573 ± 0.225	0.561 ± 0.323
	DACat	0.456 ± 0.246	0.502 ± 0.277	0.535 ± 0.383
	CAM	$\mathbf{0.223 \pm 0.209}$	0.526 ± 0.413	$\mathbf{0.843 \pm 0.240}$

8 Conclusion

In this paper, we introduced a new CCRL task to test the causal reasoning capabilities of RL agents under the contextual setting. While our proposed environment is simple and computationally efficient, we still observe that SOTA agents fail to generalize in this setup. We show that by incorporating the proposed *Contextual Attention Module*, the causal reasoning performance of the agents can be considerably improved. This module combines the context variable given by disentangled features, with the visual features from the states, and allows the agent to test interventions and discover the causal factors of the context variable. Furthermore, it generalises better to unseen contexts compared to existing baselines from the literature. As evidenced by our results, there seems to be a considerable performance drop when reasoning is required, and current SOTA shows lacking capabilities in this aspect. In our future work, we will focus on improving the efficiency of finding feasible and useful interventions, and designing mechanisms allowing an agent to ignore non-causal factors; or even answer counterfactual questions.

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational Option Discovery Algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Ossama Ahmed, Frederik Träuble, Anirudh Goyal, Alexander Neitz, Manuel Wüthrich, Yoshua Bengio, Bernhard Schölkopf, and Stefan Bauer. CausalWorld: A Robotic Manipulation Benchmark for Causal Structure and Transfer Learning. *arXiv preprint arXiv:2010.04296*, 2020.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. *arXiv preprint arXiv:1811.06272*, 2018.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic Gridworld Environment for OpenAI Gym. `https://github.com/maximecb/gym-minigrid`, 2018.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is All You Need: Learning Skills without a Reward Function. *arXiv preprint arXiv:1802.06070*, 2018.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent Independent Mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

Karol Gregor, Danilo Jimenez Rezende, and Daan Wierstra. Variational Intrinsic Control. *arXiv preprint arXiv:1611.07507*, 2016.

David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov Decision Processes. *arXiv preprint arXiv:1502.02259*, 2015.

Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E. Schapire. Contextual Decision Processes with low Bellman rank are PAC-Learnable. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Diederik Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. of the 3rd International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR)*, 2014.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proc. of the 30th International Conference on Machine Learning (ICML)*, 2013.

Vinod Nair and Geoffrey E Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-Supervised Exploration via Disagreement. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019.

Judea Pearl. The Do-Calculus Revisited. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 2012.

Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards Causal Representation Learning. *arXiv preprint arXiv:2102.11107*, 2021.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *Proc. of the 4th International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *Proc. of the 36th International Conference on Machine Learning (ICML)*, 2019.

A APPENDIX

A.1 ARCHITECTURES AND HYPERPARAMETERS

All PPO agents are trained using Adam optimizer (Kingma & Ba, 2015) with a learning rate of 7^{-4}, generalized advantage estimation (GAE) (Schulman et al., 2016) with $\lambda = 0.95$ and a discounting factor γ of 0.99. The clipping parameter ϵ is set to 0.2 and the entropy and value coefficients are set to 0.01 and 0.5, respectively. We use 8 parallel actors and updates are performed every 4096 time steps (512 per actor) with a mini-batch size of 128 and 4 update-iterations. Overall the agents are trained for 1000 epochs.

State Encoder: The state encoding network consists of four 4×4 convolutional layers with 16, 16, 32 and 32 channels and a stride of 2. Except for models relying on an attention mechanism (*SA* and *CAM*), the output of the last convolutional layer is flattened and processed by a fully-connected layer of size 512. ReLU activation function (Nair & Hinton, 2010) is applied between all layers.

Context Encoder: If applicable, agents use a separate context encoding network which is the encoder of a Convolutional Variational Autoencoder (VAE) as described in Appendix A.2. This encoder is pre-trained and not updated during training of the RL agents.

Policy and Value Networks: The policy and value networks use a fully-connected layer of size 32, ReLU activation and a linear output layer of size 3 for the policy and 1 for the value function. State and context encoder are shared across both.

A.2 VAE TRAINING AND EXAMPLES

A Convolutional Variational Autoencoder (VAE) (Kingma & Welling, 2014) architecture is used with four convolutional layers with channels $32, 32, 64, 64$, kernels $8, 4, 3, 3$, and strides $4, 2, 1, 1$, and no padding for the encoder. The decoder uses the reversed architecture of the encoder, with transposed convolutional layers instead of convolutional layers. The *Leaky Rectified Linear Units* (Maas et al., 2013) with a negative slope of 0.002 were used as the non-linearity for the hidden layers of the VAE. The decoder uses a $TanH$ output activation, and all input data was normalized to $[-1, 1]$, whenever fed to the VAE. The model was trained using Adam optimizer with learning rate of 0.0003 and batch-size of 128, for 100 epochs, on a dataset comprised of 50k random samples from the environment, which were collected and used as the training data, containing both $training$ and $testing$ colours. As the objective, we use the annealing VAE proposed in (Burgess et al., 2018). Random samples from the dataset, and their reconstructions using our fully trained VAE are provided in Figure 5.

A.3 DEFINED CONTEXTS

For training, we use a set of 4 context pairs as shown in Table 2. For evaluation, we have two scenarios, *seen-colours* and *unseen-colours*.

In the *seen-colours* scenario, we use those colours that were already used during training for goodie, agent and obstacle and create all possible combinations (minus the ones used for training), resulting in 48 new contexts. As for training, we partition those with the same colours into pairs. The goal colour is arbitrarily chosen for each pair to be either *plum*, *maroon*, *rosy brown* or *lavender*.

For the *unseen-colour* scenario, we introduce eight new colours *khaki*, *pink*, *dark olive green*, *pale violet red*, *yellow*, *purple*, *orange* and *green* and again create all possible combinations for obstacle and goodie/agent. As before, contexts are partitioned into pairs based on the same colours and the goal colour is arbitrarily chosen for each pair using the same goal colours as in the *seen-colours* scenario.

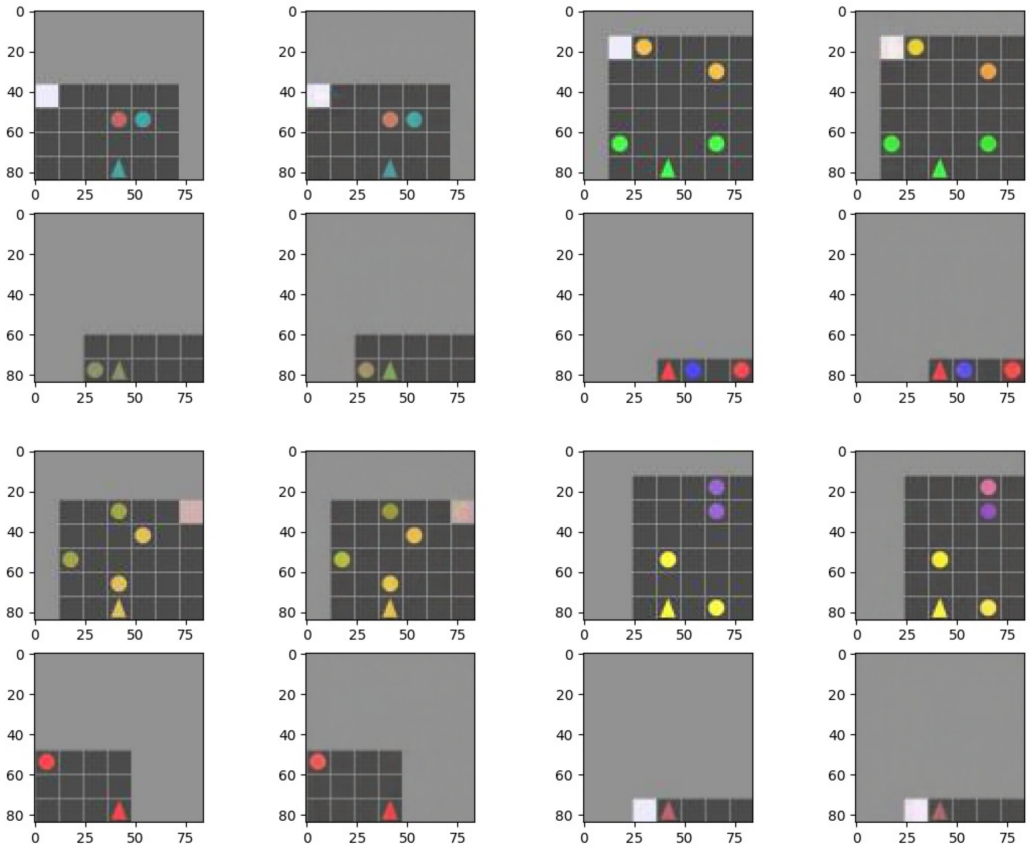

Figure 5: Reconstructed samples from the annealed VAE used in our experiments. The first and third columns are original inputs, and the second and fourth columns are their reconstruction.

Table 2: Causal Context configurations for training.

Context Pair	Context	Goodie/Agent	Obstacle	Goal
1	1	red	blue	plum
	2	blue	red	plum
2	3	navy	cyan	maroon
	4	cyan	navy	maroon
3	5	olive	gold	rosy brown
	6	gold	olive	rosy brown
4	7	brown	teal	lavender
	8	teal	brown	lavender