GalilAI: Out-of-Task Distribution Detection using Causal Active Experimentation for Safe Transfer RL

Sumedh A Sontakke[*] University of Southern California
Stephen Iota[*] University of Southern California
Zizhao Hu[*] University of Southern California
Arash Mehrjou MPI for Intelligent Systems
Laurent Itti University of Southern California
Bernhard Schölkopf MPI for Intelligent Systems

Abstract

Out-of-distribution (OOD) detection is a well-studied topic in supervised learning. Extending the successes in supervised learning methods to the reinforcement learning (RL) setting, however, is difficult due to the data generating process — RL agents actively query their environment for data, and the data are a function of the policy followed by the agent. An agent could thus neglect a shift in the environment if its policy did not lead it to explore the aspect of the environment that shifted. Therefore, to achieve safe and robust generalization in RL, there exists an unmet need for OOD detection through active experimentation. Here, we attempt to bridge this lacuna by first defining a causal framework for OOD scenarios or environments encountered by RL agents in the wild. Then, we propose a novel task: that of Out-of-Task Distribution (OOTD) detection. We introduce an RL agent that actively experiments in a test environment and subsequently concludes whether it is OOTD or not. We name our method GalilAI, in honor of Galileo Galilei, as it discovers, among other causal processes, that gravitational acceleration is independent of the mass of a body. Finally, we propose a simple probabilistic neural network baseline for comparison, which extends extant Model-Based RL. We find that GalilAI outperforms the baseline significantly. See visualizations of our method `here`.

[*]These authors contributed equally to this work

Proceedings of the 25[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

1 Introduction and Related Work

Generalization to near-distribution shifts caused by natural perturbations and **Detection** of out-of-distribution shifts caused by artificial perturbations (adversarial attacks) are central desiderata of modern decision-making systems. Significant advances have been made in supervised learning systems on both fronts - with work in transfer/meta-learning aiding the ability of ML systems to generalize across shifts in input distributions [Schmidhuber, 2007, Santoro et al., 2016, Finn et al., 2017]. Such methods learn internal representations which are invariant to perturbations occurring in data [Bengio, 2013]. These invariant representations are subsequently used for domain adaptation [Zhao et al., 2019, Muandet et al., 2013], with applications in music [Blumensath and Davies, 2005] and speech [Serdyuk et al., 2016]. Out-of-distribution Detection for the supervised learning domain has also made significant advances [Hendrycks and Gimpel, 2016, DeVries and Taylor, 2018, Liang et al., 2017, Goodfellow et al., 2014], with the development of both training-time methods [Xiao et al., 2020] (alterations to typical supervised training to make models robust to OOD inputs) and inference-time methods (utilizing the features of a fully trained model to detect OOD samples)[Hsu et al., 2020].

While attempts have been made in generalization in the space of sequential long-horizon RL and decision-making [Finn et al., 2017, Nagabandi et al., 2018, Gupta et al., 2018, Parisotto et al., 2015, Rakelly et al., 2019, Zintgraf et al., 2019], Out-of-Distribution Detection is fairly unexplored. To our knowledge, our work is the first that offers a concrete causal framework for OOD Detection.

We motivate the need for OOD Detection in RL with an example. Consider an agent that has learnt to land an aircraft for various values of directions and velocity of crosswinds. Now consider the situation when one of the airplane's engines fails when the agent is deployed. Current RL systems would assume that the observations they receive from this test environment were caused by perhaps high crosswinds and would subsequently increase fuel flow to the engines - a potentially disastrous strategy. On the contrary, a seasoned pilot might perform an experiment - perhaps yawing the aircraft from side-to-side, concluding that due to the low controllability of the aircraft, the engine was somehow compromised. Our work extends that of Sontakke et al. by utilizing advances in algorithmic information theory and curiosity-based reinforcement learning to "encourage" the RL agent to perform such experimental behaviors and conclude whether a test-time environment is out-of-training-distribution or not.

During our experiments, we find that our agent discovers the **Galilean Equivalence Principle**, managing to successfully decouple the effect of mass and gravitational acceleration. For this reason, we refer to the agent as GalilAI (pronounced Galilei). The contributions of our work are as follows:

- **Causal transfer:** We offer a causal perspective on transfer learning in RL and provide a theoretical framework for defining various classes of transfer RL problems.
- **Causal active experimentation (GalilAI) for safe transfer RL:** We extend the work of Sontakke et al. to provide an algorithm aimed at improving the safety of transfer reinforcement learning by detecting whether a given test environment is out-of-distribution or not. If an environment is detected as OOD, the agent could relinquish control of a system to a human operator [Amodei et al., 2016].
- **Probabilistic baseline:** Due to a lack of prior work in the field, we propose a simple probabilistic neural network baseline for OOD Detection of environments in RL. We compare GalilAI and the PNN in complex robotic domains such as the Causal World [Ahmed et al., 2020] and Mujoco [Todorov et al., 2012a].

2 Preliminaries

Definition 1 (Causal factors) *Consider the POMDP ($\mathcal{O}$, $\mathcal{S}$, $\mathcal{A}$, ϕ, θ, r) with observation space $\mathcal{O}$, state space $\mathcal{S}$, action space $\mathcal{A}$, the transition function ϕ, emission function θ, and the reward function r. Let $\mathbf{o}_{0:T} \in \mathcal{O}^T$ denote a trajectory of observations of length T. Let $d(\cdot,\cdot) : \mathcal{O}^T \times \mathcal{O}^T \to R_+$ be a distance function defined on the space of trajectories of length T. The set $H = \{\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{K-1}\}$ is called a set of $\epsilon-$causal factors if for every $\mathbf{h}_j \in H$, there exists a unique sequence of actions $\mathbf{a}_{0:T}$ that clusters the observation trajectories into m disjoint sets $C_{1:m}$ such that $\forall C_a, C_b$, a minimum separation distance of ϵ is ensured:*

$$\min\{d(\mathbf{o}_{0:T}, \mathbf{o}'_{0:T}) : \mathbf{o}_{0:T} \in C_a, \mathbf{o}'_{0:T} \in C_b\} > \epsilon \quad (1)$$

and that $\mathbf{h}_j$ is the cause of the obtained trajectory of states i.e. $\forall v \neq v'$,

$$p(\mathbf{o}_{0:T}|do(\mathbf{h}_j = v), \mathbf{a}_{0:T}) \neq p(\mathbf{o}_{0:T}|do(\mathbf{h}_j = v'), \mathbf{a}_{0:T}) \quad (2)$$

where $do(\mathbf{h}_j)$ corresponds to an intervention on the value of the causal factor $\mathbf{h}_j$.

According to Definition 1, a causal factor $\mathbf{h}_j$ is a variable in the environment the value of which, when intervened on (i.e., varied) using $do(\mathbf{h}_j)$ over a set of values, results in trajectories of observations that are divisible into disjoint clusters $C_{1:m}$ under a particular sequence of actions $\mathbf{a}_{0:T}$. These clusters represent the quantized values of the causal factor. For example, mass, which is a causal factor of a body, under an action sequence of a grasping and lifting motion with fixed force, may result in 2 clusters, liftable (low mass) and not-liftable (high mass).

2.1 POMDPs and Causal POMDPs

Classical POMDPs ($\mathcal{O}$, $\mathcal{S}$, $\mathcal{A}$, ϕ, θ, r) consist of an observation space $\mathcal{O}$, state space $\mathcal{S}$, action space $\mathcal{A}$, the transition function ϕ, emission function θ, and the reward function r. An agent in an unobserved state $\mathbf{s}_t$ takes an action $\mathbf{a}_t$ and consequently causes a transition in the environment through $\phi(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$. The agent receives an observation $\mathbf{o}_{t+1} = \theta(\mathbf{s}_{t+1})$ and a reward $\mathbf{r}_{t+1} = r(\mathbf{s}_t, \mathbf{a}_t)$. **Causal POMDPs** explicitly model the effects of causal factors on the transition and emission functions by dividing the state into the controllable state $\mathbf{s}^c_t$ and the causal factor, $\mathcal{H}$. The causal factors of an environment cannot be manipulated by the agent, but their values affect the outcome of an action taken by the agent. Thus the transition function of the controllable state is:

$$\phi(\mathbf{s}^c_{t+1}|\mathbf{s}^c_t, f_{sel}(\mathcal{H}, \mathbf{s}^c_t, \mathbf{a}_t), \mathbf{a}_t) \quad (3)$$

where f_{sel} is the implicit Causal Selector Function which selects the subset of causal factors affecting the transition defined as:

$$f_{sel} : \mathcal{H} \times \mathcal{S} \times \mathcal{A} \to \wp(\mathcal{H}) \quad (4)$$

where $\wp(\mathcal{H})$ is power-set of $\mathcal{H}$ and $f_{sel}(\mathcal{H}, \mathbf{s}_t^c, \mathbf{a}_t) \subset \mathcal{H}$ is the set of effective causal factors for the transition $\mathbf{s}_t \to \mathbf{s}_{t+1}$ i.e., $\forall v \neq v'$ and $\forall \mathbf{h}_j \in f_{sel}(\mathcal{H}, \mathbf{s}_t^c, \mathbf{a}_t)$:

$$\phi(\mathbf{s}_{t+1}^c | do(\mathbf{h}_j = v), \mathbf{s}_t^c, \mathbf{a}_t) \neq \phi(\mathbf{s}_{t+1}^c | do(\mathbf{h}_j = v'), \mathbf{s}_t^c, \mathbf{a}_t) \tag{5}$$

where $do(\mathbf{h}_j)$ corresponds to an external intervention on the factor $\mathbf{h}_j$ in an environment.

Intuitively, this means that if an agent takes an action $\mathbf{a}_t$ in the controllable state $\mathbf{s}_t^c$, the transition to $\mathbf{s}_{t+1}^c$ is caused by a subset of the causal factors $f_{sel}(\mathcal{H}, \mathbf{s}_t^c, \mathbf{a}_t)$. For example, if a body on the ground (i.e., state $\mathbf{s}_t^c$) is thrown upwards (i.e., action $\mathbf{a}_t$), the outcome $\mathbf{s}_{t+1}$ is caused by the causal factor gravity (i.e., $f_{sel}(\mathcal{H}, \mathbf{s}_t^c, \mathbf{a}_t) = \{\text{gravity}\}$), a singleton subset of the global set of causal factors. The $do()$ notation expresses this causation. If an external intervention on a causal factor is performed, e.g., if somehow the value of gravity was changed from v to v', the outcome of throwing the body up from the ground, $\mathbf{s}_{t+1}$, would be different.

2.2 Algorithmic Information Theoretic View on Causality

Causality can be motivated from the perspective of algorithmic information theory [Janzing and Schölkopf, 2010]. Consider the Gated Directed Acyclic Graph of the observed variable $\mathbf{O}$ and its causal parents (Figure 1). Each causal factor has its own causal mechanism, jointly bringing about $\mathbf{O}$. The action sequence $\mathbf{a}_{0:T}$ serves a gating mechanism, allowing or blocking particular edges of the causal graph using the implicit Causal Selector Function (Equation (4)). A central assumption of our approach is that causal factors are independent, i.e., the Independent Mechanisms Assumption [Schölkopf et al., 2012, Parascandolo et al., 2018, Schölkopf, 2019]. The information in $\mathbf{O}$ is then the sum of information "injected" into it from the multiple causes, since, loosely speaking, for information to cancel, the mechanisms would need to be algorithmically dependent [Janzing and Schölkopf, 2010]. Thus, the information content in $\mathbf{O}$ will be greater for a larger number of independent causal parents in the graph.

$$L(\mathbf{O}) \propto |PA(\mathbf{O}))| \tag{6}$$

where $L(\cdot)$ is the Minimum Description Length (MDL), a tractable substitute of the Kolmogorov Complexity [Rissanen, 1978, Grunwald, 2004]).

2.3 Causal Curiosity

Causal curiosity [Sontakke et al., 2021] allows an RL agent to discover sequences of actions that bring out the effect of a single causal factor while ignoring the effects of all other. This is similar to how a human scientist studying multiple mechanisms in their environment would behave whilst following the One-Factor-at-a-Time (OFAT) paradigm of experiment design [Fisher, 1936]. For e.g., when interacting with objects of varying mass and shape, a human scientist will learn a perfect lifting sequence that grasps all shapes and then use it to test out the mass of each object.

Thus, *Causal Curiosity selects one among multiple competing causal mechanisms* and generates a sequence of actions that bring out the effect of the selected mechanism. This is done by attempting to learn a simple model of the environment with capacity low enough to learn about only a single causal mechanism at a time. One could conceive of this by assuming that the generative model for $\mathbf{O}$, $\mathbf{M}$ has low Kolmogorov Complexity. A low capacity bi-modal model is assumed. The Minimum Description Length (MDL), $L(\cdot)$ is utilized as a tractable substitute of the Kolmogorov Complexity Rissanen [1978], Grunwald [2004]). Subsequently, the following optimization problem is solved.

$$\mathbf{a}_{0:T}^* = \arg\min_{\mathbf{a}_{0:T}} (L(\mathbf{M}) + L(\mathbf{O}|\mathbf{M})) \tag{7}$$

where each observed trajectory $\mathbf{O} = \mathbf{O}(\mathbf{a}_{0:T})$ is a function of the action sequence. Thus the resulting action sequence from the optimization in Equation (7) will result in an action sequence that brings out the effect of a single caual factor. Having established this, we now introduce a causal perspective on transfer.

2.4 Causal Perspective on Transfer

Consider the set of POMDPs $P = \{\mathbf{p}_0, \mathbf{p}_1, \ldots\}$ parameterized by the tuple ($\mathcal{O}$, $\mathcal{S}$, $\mathcal{A}$, ϕ, θ, r, $H' \subset H$) with observation space $\mathcal{O}$, state space $\mathcal{S}$, action space $\mathcal{A}$, the transition function ϕ, emission function θ, and the reward function r, with the set of causal factors $H' \subset H$, i.e., subset of the global causal factors, varied over a range of values and the remaining $H - H'$ held constant.

Definition 2 (In-Task-Distribution Transfer) *An in-task-distribution transfer occurs when an agent trained on P is launched into a POMDP $\mathbf{p}'$ where $\forall$ $\mathbf{h} \in H - H'$ the values assumed by $\mathbf{h}$ remain unchanged (assume the same values as in P).*

Definition 3 (Out-of-Task-Distribution Transfer) *An Out-of-Task-distribution transfer occurs when an agent trained on P is launched into a POMDP $\mathbf{p}'$ where $\exists$ $\mathbf{h} \in H - H'$ which assumes a value different from the value it had in P.*

Consider a transfer learning agent training to lift cubes with varying masses and sizes, i.e., $H' = \{mass, size\}$.

An In-Task-Distribution Transfer scenario occurs if at test-time it encounters an cube of an unseen mass/size combination. An Out-of-Task-Distribution scenario occurs if it is required to lift a cube with a broken actuator. This is because the causal factor *actuator* $\in H - H'$ which was held constant during training (agent trained is using a healthy actuator) is required to lift using a broken actuator at test-time. We would like to be able to detect such faults while generalizing to known causal factors.

3 Method

Setup We consider the scenario where a learning agent is trained on a set of POMDPs $P = \{\mathbf{p}_0, \mathbf{p}_1, \ldots\}$ parameterized by the tuple ($\mathcal{O}$, $\mathcal{S}$, $\mathcal{A}$, ϕ, θ, r, $H' \subset H$) with observation space $\mathcal{O}$, state space $\mathcal{S}$, action space $\mathcal{A}$, the transition function ϕ, emission function θ, and the reward function r, with the set of causal factors $H' \subset H$, i.e., subset of the global causal factors, varied over a range of values and the remaining $H - H'$ held constant.

We assume that the learning agent is able to learn $\mathbf{z} = Z_\phi(\mathbf{p})$, called *belief function*, using each of the training environments which generates a representation for the intervened causal factors, i.e., $H' \subset H$. This assumption is quite general - the RL systems that are capable of performing well over different environments can be assumed to either explicitly model such representations (as in [Rakelly et al., 2019, Zintgraf et al., 2019, Perez et al., 2020]) or implicitly (as in [Finn et al., 2017, Nagabandi et al., 2018]). At test time, the agent is launched into a novel environment $\mathbf{p}'$ which is either an In-Task-Distribution Transfer (see Definition 2) or Out-of-Task-Distribution Transfer (see Definition 3).

3.1 Construction of the Belief Set

The agent performs inference in the novel test environment $\mathbf{p}'$ using $Z_\phi(\mathbf{p}')$. We assume that the agent has access to $\{Z_\phi(\mathbf{p}) : \mathbf{p} \in P\}$, i.e., the belief representation for the training environments. The agent then collects all training environments that lie near $\mathbf{p}'$ in the space of the learned belief functions into the ball $\mathcal{B}$ called the *belief set* defined as,

$$\mathcal{B} := \{\mathbf{p}_i : d(q_\phi(\mathbf{z}|\mathbf{p}')||q_\phi(\mathbf{z}|\mathbf{p}_i)) < \epsilon\} \tag{8}$$

where $d(\cdot|\cdot)$ is a distance function (e.g., Euclidean) in the latent space and ϵ is a design hyperparameter.

Thus, for example, in a lifting task of cubes of various masses, if the agent fails to lift a cube at test time, it constructs the belief ball consisting of the training environments with close representations, i.e., heavy cubes and adds them to the belief set $\mathcal{B}$. Depending on the cause for the failure of the agent in lifting the cube, the situation goes into one of the following branches: (1) The test environment requires an **In–Task-Distribution Transfer**, i.e., the test-time block is actually a heavy block or (2) The test environment requires **Out-of-Task-Distribution Transfer**, i.e., a broken actuator makes a light block seem heavy.

3.2 Belief Verification

Subsequently, the agent optimizes causal curiosity on $\mathcal{B} \cup \{\mathbf{p}'\}$. As in Equation (7), a low capacity binary clustering model is considered. Thus, the following optimization procedure is implemented:

$$\begin{aligned}\arg\max_{\mathbf{a}_{0:T} \in \mathcal{A}^T} [\min\{d(\mathbf{o}_{0:T}, \mathbf{o'}_{0:T}) : \mathbf{o}_{0:T} \in C_1, \mathbf{o'}_{0:T} \in C_2\} - \\ \max\{d(\mathbf{o}_{0:T}, \mathbf{o''}_{0:T}) : \mathbf{o''}_{0:T}, \mathbf{o}_{0:T} \in C_1\} - \\ \max\{d(\mathbf{o'}_{0:T}, \mathbf{o'''}_{0:T}) : \mathbf{o'}_{0:T}, \mathbf{o'''}_{0:T} \in C_2\}]\end{aligned} \tag{9}$$

where $\mathbf{O}$ is the observation obtained by applying action sequence $\mathbf{a}_{0:T}$. Clusters C_1 and C_2 represent the bimodal model.

In-Task Distribution If the test environment $\mathbf{p}'$ is In-Task Distribution, then the variance of values assumed by the causal factors H' in the set of environments $\mathcal{B} \cup \{\mathbf{p}'\}$ is small and the clusters are *not well-separated*. Thus optimizing causal curiosity as in Equation (9) will produce action sequences that result in observations that cluster in a distributed manner as in pane **A** of Figure 2.

Intuitively, if the agent has learnt to interact with blocks of various masses and at test time is presented with a heavy block, the outcome of its interaction with the test block (i.e., $\mathbf{p}'$) will not differ significantly in comparison with the heavy blocks it interacted with during training.

Out-of-Task-Distribution However, during the optimization of Equation (9) in the OOTD case, 2 competing causal mechanisms will exist - one induced by the set H' and the other from the set $H - H'$. The mechanism caused by $H - H'$ will however dominate as all environments in $\mathcal{B}$ will have the same values for $H - H'$ while $\mathbf{p}'$ will have a different value. Thus, the resulting clusters for the causal mechanism from $H - H'$ will be *well-separated*. Subsequently, the causal curiosity reward (Equation (9)) will be higher for selecting the causal mechanism induced by $H - H'$.

Intuitively, as in the above example of an agent interacting with blocks of varying masses but constant size ξ, if at test time, the agent is provided with a block of

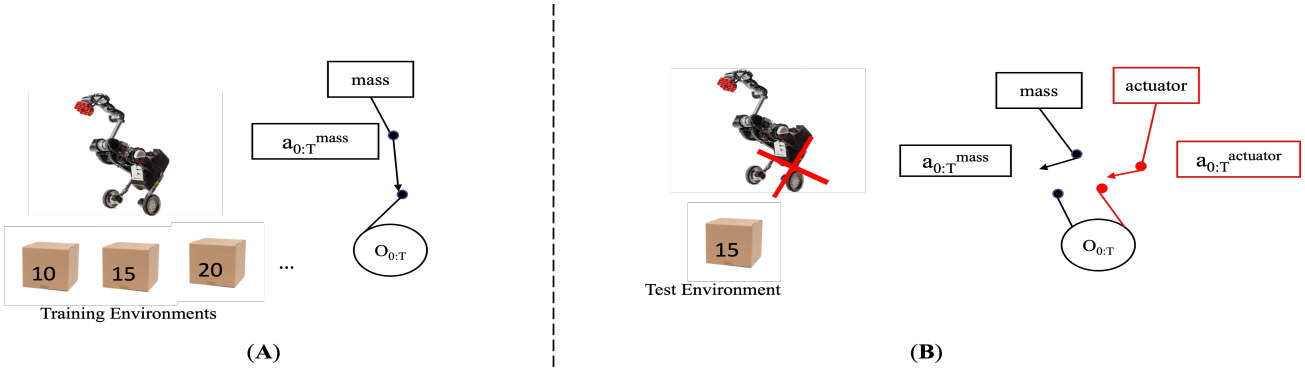

Figure 1: **Out-of-Task Distribution Transfer.** Pane **A** shows a training-time scenario where an agent learns to interact with environments containing objects for varying values of mass. The causal graph is gated as particular action sequences either obfuscate or underscore the effects of certain causal factors. Pane **B** represents the inference-time scenario where the a causal factor, actuator health, held constant during training is varied at inference.

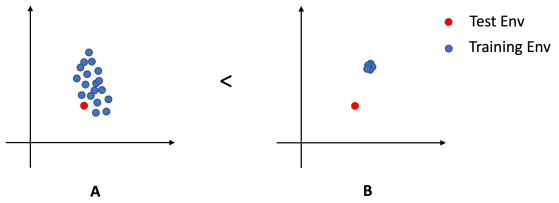

Figure 2: **Visualization of the Observation O.** Pane **A** represents the observation variables obtained after the optimization of Equation (9) during an In-Task-Distribution Transfer. Causal Curiosity will be quite low in such a case as the bi-modal clustering would be poor. Pane **B** represents the case when OOTD transfer occurs - the causal curiosity reward would be high as the bi-modal clustering would be near-perfect.

low mass and a new size $\xi' \neq \xi$, the causal curiosity reward for the size mechanism will be higher because a perfect binary clustering is possible (as in pane **B** in Figure 2) where one cluster contains observations from training environments (blue cluster) corresponding to size s while the other cluster corresponds to the test environment with size s' (red cluster). Thus, if the test environment $\mathbf{p}'$ lies in its own cluster after optimizing causal curiosity on $\mathcal{B} \cup \{\mathbf{p}'\}$, then GalilAI concludes $\mathbf{p}'$ to be OOTD, i.e.,

$$\text{Is\_OOTD}(p') = \begin{cases} 1, & \text{if } \mathbf{p}' \text{ lies in its own cluster} \\ 0, & \text{otherwise} \end{cases}$$

Note, the causal curiosity for a known causal factor, (In-Task-Distribution Transfer) will be less than the causal curiosity for an unknown causal factor (OOTD Transfer) as seen in Figure 2.

3.3 Probabilistic Baselines

A natural extension of Model-based learning methods for OOTD is possible. We question whether such an extension yields good results. We test whether OOTD Detection is possible by simply learning a model of the training and test environments and using using the discrepancy of the outputs to detect whether the learnt test-time model represents a task from OOTD.

We utilize an ensemble of probabilistic neural networks (PNNs) [Lakshminarayanan et al., Chua et al., 2018], which is a generative neural network whose output neurons parameterize a probability distribution $p_\theta(y|\mathbf{x})$; a mean value corresponds to the believed label $\hat{y}$ along with some degree of uncertainty θ.

For an environment $\mathbf{p}$, we estimate the environment transition function $\phi_{\mathbf{p}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$ using an ensemble of PNNs $f_\phi^{\mathbf{p}}(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t))$. We are interested in the disagreement between a novel test environment $\mathbf{p}'$relative to a training environment $\mathbf{p}$ which we measure using the relative entropy between $f_\phi^{\mathbf{p}'}$ and $f_\phi^{\mathbf{p}}$ given by $KL(f_\phi^{\mathbf{p}'}||f_\phi^{\mathbf{p}})$:

$$D_{KL}(f_\phi^{\mathbf{p}'}||f_\phi^{\mathbf{p}}) = \frac{1}{2}[\log \frac{|\Sigma(f_\phi^{\mathbf{p}})|}{|\Sigma(f_\phi^{\mathbf{p}'})|} - k + (\mu(f_\phi^{\mathbf{p}'}) - \mu(f_\phi^{\mathbf{p}'}))^T \Sigma^{-1}(f_\phi^{\mathbf{p}})(\mu(f_\phi^{\mathbf{p}'}) - \mu(f_\phi^{\mathbf{p}})) + \Sigma^{-1}(f_\phi^{\mathbf{p}})\Sigma(f_\phi^{\mathbf{p}'})] \tag{10}$$

where k is the dimensionality of the environment's observation space; $\mathbf{s}_t \in R^k$, $\Sigma(\cdot)$ is covariance and $\mu(\cdot)$ is mean and $(\cdot)$ is trace.

We use Negative Log Loss as a scoring rule for PNNs, and KL divergence as a measure of distribution dis-

agreement. We utilize a thresholding technique here to detect OOTD. We train $f_\phi^{\mathbf{p}'}$ and $f_\phi^{\mathbf{p}}$ over multiple random seeds and use the mean $KL(f_\phi^{\mathbf{p}'}||f_\phi^{\mathbf{p}})$ of the first k seeds as the threshold. Detection is then performed using:

$$\text{Is\_OOTD}(p') = \begin{cases} 1, & \text{if } KL(f_\phi^{\mathbf{p}'}||f_\phi^{\mathbf{p}}) > threshold \\ 0, & \text{otherwise} \end{cases}$$

4 Experiments

Our work has 2 main thrusts - the discovered *experimental behaviors* and the *Out-of-Task-Distribution Detection* obtained from the outcome of the behaviors in environments. We visualize these learnt behaviors and verify that they are indeed semantically meaningful and interpretable. We quantify the utility of the learnt behaviors to perform OOTD detection.
Causal World. We use the Causal World Simulation [Ahmed et al., 2020] based on the Pybullet Physics engine to test our approach. The simulator consists of a 3-fingered robot, with 3 joints on each finger. We constrain each environment to consist of a single object that the agent can interact with. The causal factors that we manipulate for each of the objects are size and mass of the blocks and the damping factor and control frequency of the robotic motors. The simulator allows us to capture and track the positions and velocities of each of the movable objects in an environment.
Mujoco Control Suite. We also perform OOTD Detection on 3 articulated agents that try to learn locomotion - Half-Cheetah, Hopper, and Walker. For each agent type, the causal factors that we intervene on include the mass of the robot, and wind and gravity in the environment, and the friction between the robot actuators and the ground.

4.1 Generalized Experimental Setup

To test our approach, we train a transfer RL algorithm - in our case, Causal Curiosity [Sontakke et al., 2021] on multiple environments with causal factor A assuming values $A = a \in A$ where A is a set of values causal factor A can assume. For example, we train an agent to interact with blocks of varying masses (here mass is causal factor A). At test time, we generate a range of values of a causal factor B previously held fixed. Thus, in the above example, having been trained on varying values of mass, we now generate a range of values for the control frequency (causal factor B) of the robot actuators (previously held constant at some η). For each pair of values of (B, A) causal factors, we report the accuracy of detection over 10 random seed experiments. Thus, in the above example, all environments with $(control = \eta, mass = m)$ are considered In-Task-Distribution Transfer while all others are Out-of-Task Distribution.
Interpreting Results For each pair of test-time and training-time causal factors, we vary both over a range of values. Consider for example Figure 3, where we vary the mass of the blocks the robot interacts with during training and at test-time, it receives environments with a different perception frequency. During training in Figure 3 Pane **C**, the perception frequency was at 1 (corresponding to the column at Perception Frequency = 1). For each pair of (Control Frequency, Mass), we run our method over 10 random seeds. The value at each (x, y) position corresponds to the number of times during the 10 runs, GalilAI concluded that the test environment was OOTD. Figure 3 Pane **C** is an example of perfect detection - no false positives (column above $x = 1$ is zero) and 100% detection when Perception Frequency is varied. Other experiments depict varying degrees of detection success.

4.2 Causal World Experiments

During training, we vary either the mass or size of the block in an environment. At test time, the agent interacts with an environment with 3 possible errors - **(1) Distributional Shift of the Environment**: Change in the physical features of the block (2) **Perception Defect**: Frequency of perception changes (i.e., framerate of sensors) which affects the perception-to-action loop and **(3) Actuator Defect**: the damping coefficient of the arm actuators is varied, which affects the dynamics of the robotic arm.

Figure 3 depicts experiments with mass and size of blocks as training-time causal factors and Perception Frequency, Damping Factor and Mass as test-time causal factors. Each of these experiments yield no false positives as the columns above in the fixed value of the test-time causal factor have zero detections. Figure 3 Pane **C** shows the agent has a perfect detection performance in detecting the perception defect. Figure 3 panes **A** and **B** show that detection is successful when the test-time value of the unseen causal factor is some distance away from its constant value during training. The agent is more likely to detect an unknown causal factor when we make a larger change to it (larger values near the left and right border).

4.3 Mujoco Experiments

We perform experiments with the Mujoco control suite [Todorov et al., 2012b] as well. During training, we manipulate the mass of the friction and the friction coefficients between the agent actuators and ground. At test time, we manipulate the wind and gravity in the environment and the mass of the agent. The in-

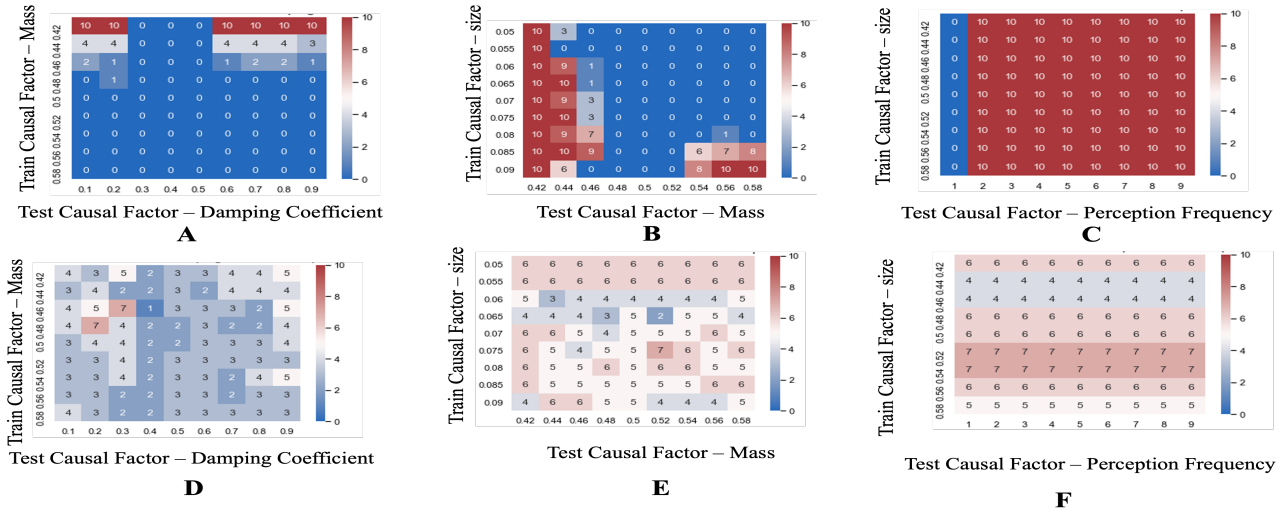

Figure 3: **Causal World experiments.** Subfigures **A** – **C** refer to GalilAI, and subfigures **D** – **F** refer to the probabilistic baseline. Each (x, y) pair on the plot corresponds to an $(unseen, seen)$ pair of causal factors. The value at each (x, y) pair depicts the performance of each method across measure performance as the number of correctly classified environments on 10 different random seeds. In-distribution value of mass is 0.5; Damping Coefficient is 0.5; Perception frequency is 1.0. Ideally, the column above the training value of the OOTD causal factor should be 0, while all other columns should be 10 as is the case in Pane **C**.

distribution value of wind is 0.0, gravity is -9.8 and mass is 1.0. Discerning wind while being invariant to agent mass (Panes **A** and **D** in each sub-figure of Figure 4) is a relatively easy endeavour with the half-cheetah resulting in the highest accuracy across each of the random seeds. The task of discerning mass while being invariant to friction also yields high accuracy of detection when the test mass varies significantly in comparison with training time mass (red columns on right and left edges of Panes **C** and **F** in Figure 4). However, it suffers from poor detection at $0.8\times$ and $1.2\times$ the default mass for cheetah and hopper. The hardest task is that of discerning gravity while being invariant to mass - a task that requires discovering the **Galilean Equivalence Principle**, i.e., that the acceleration due to gravity is independent of mass. While the success of GalilAI is limited when gravity is in the vicinity of 9.8, it begins to successfully learn to detect changes in gravity as it deviates from 9.8.

4.4 Interpretation of Learned Behaviours

For visualizations of our method, see **here.** We analyze whether the discovered experimental behaviors are actually semantically meaningful. We find that the agent is able to discover many semantically meaningful behaviors that underscore the effect of a new causal factor previously held constant during training. Chiefly, we find that the 17th century philosopher Galileo Galilei and his namesake GalilAI agree that mass and gravitational acceleration are decoupled - GalilAI learns a free-falling behavior that mimics Galileo's experiments of dropping objects to discern the gravity of an environment whilst being invariant to the agent mass.

In Mujoco Experiments, for mass as training time causal factor and wind as test-time factor, the agent learnt to use its body as a sail and allow the wind to carry it along. It also learnt to do front-flips and rolls in the direction of the wind, using the wind to help it along. For friction as training causal factor and mass as the test time causal factor, the agent also learnt to perform headstands to test out its mass while avoiding any horizontal locomotion allowing it to be invariant to the friction coefficients.

In Causal World Experiments, for size as training time causal factor and mass as test-time factor, the agent learnt a relay-kick action when one of the finger push the object to the other finger, who makes a further push on it. This relay can only be finished on small mass blocks, thus distinguishing the causal factor.

5 Conclusion

In this work, we propose a novel task - that of Out-of-Task Distribution (OOTD) Detection and offer a causally inspired solution for the same. We find that simplistic extensions of existing model-based methods result in suboptimal performance with either low-

detection accuracy and high false positive rate. We show the efficacy of our method in both a variety of embodied robotic environments spanning 2 simulation engines. We find GalilAI has the ability learn complex causal mechanisms and is a first step towards safer transfer/meta-RL.

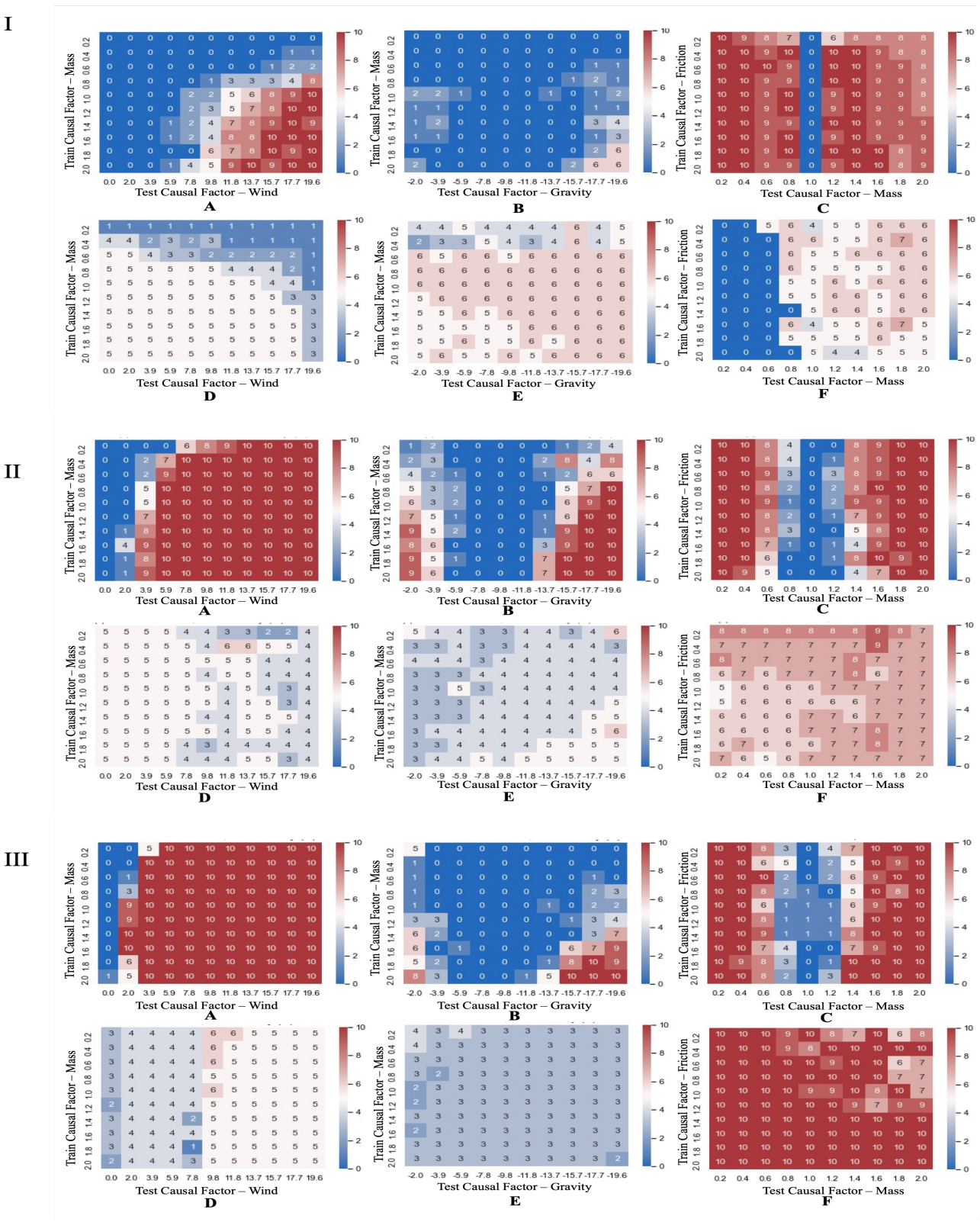

Figure 4: **Mujoco experiments.** Plots **I, II** and **III** correspond to Hopper, Walker and Cheetah environments respectively. Within each, subfigures **A** – **C** refer to GalilAI, and subfigures **D** – **F** refer to the probabilistic baseline. Each (x, y) pair on the plot corresponds to an $(unseen, seen)$ pair of causal factors. The value at each (x, y) pair depicts the performance of each method across measure performance as the number of correctly classified environments on 10 different random seeds. In-distribution value of wind is 0.0; gravity is -9.8; mass is 1.0.

References

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.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Yoshua Bengio. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer, 2013.

Thomas Blumensath and Mike Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):50–57, 2005.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *CoRR*, abs/1805.12114, 2018. URL `http://arxiv.org/abs/1805.12114`.

Terrance DeVries and Graham W Taylor. Learning confidence for out-of-distribution detection in neural networks. *arXiv preprint arXiv:1802.04865*, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

Ronald Aylmer Fisher. Design of experiments. *Br Med J*, 1(3923):554–554, 1936.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Peter Grunwald. A tutorial introduction to the minimum description length principle. *arXiv preprint math/0406077*, 2004.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. *arXiv preprint arXiv:1802.07245*, 2018.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10951–10960, 2020.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 10–18, 2013.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

Giambattista Parascandolo, Niki Kilbertus, Mateo Rojas-Carulla, and Bernhard Schölkopf. Learning independent causal mechanisms. In *International Conference on Machine Learning*, pages 4036–4044. PMLR, 2018.

Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

Christian F Perez, Felipe Petroski Such, and Theofanis Karaletsos. Generalized hidden parameter mdps: Transferable model-based rl in a handful of trials. *AAAI Conference On Artifical Intelligence*, 2020.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International conference on machine learning*, pages 5331–5340. PMLR, 2019.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

Jürgen Schmidhuber. Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence*, pages 199–226. Springer, 2007.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1255–1262, 2012.

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

Dmitriy Serdyuk, Kartik Audhkhasi, Philémon Brakel, Bhuvana Ramabhadran, Samuel Thomas, and Yoshua Bengio. Invariant representations for noisy speech recognition. *arXiv preprint arXiv:1612.01928*, 2016.

Sumedh A Sontakke, Arash Mehrjou, Laurent Itti, and Bernhard Schölkopf. Causal curiosity: Rl agents discovering self-supervised experiments for causal representation learning. In *International Conference on Machine Learning*, pages 9848–9858. PMLR, 2021.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012a. doi: 10.1109/IROS.2012.6386109.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012b.

Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *arXiv preprint arXiv:2003.02977*, 2020.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *International Conference on Machine Learning*, pages 7523–7532. PMLR, 2019.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. *arXiv preprint arXiv:1910.08348*, 2019.

A Implementation details

A.1 Planner

The Experiment Planner consists of a uniform distribution control planner with Cross Entropy Method Model Predictive Control. Each planner is initialized to a uniform distribution $\mathcal{U}(controlLow, controlHigh)$. For **Mujoco** experiments, each planner consists of 20 sampled plans per iteration. Each sampled plan consists of 6 control signals applied for a duration of 10 frames, for a total of 60 frames per episode. For **Causal World** experiments, each planner similarly consists of 20 sampled plans per iteration, with each action applied for a longer duration, for a total of 198 frames per episode. In both cases, each sampled plan is applied to each of the considered environments. At the end of each training iteration, the top 10% of plans are used to update the agent's action distribution. In total, training required 20 full iterations.

In general, during training, the agent learns a sequence of actions to maximize the Causal Curiosity reward across 9 different environments, e.g. block mass of 0.1 to 0.9 with step 0.1. The learned action sequence will group the training environments into 2 clusters, such as a large mass cluster and a small mass cluster. Then, using the action sequence which maximizes the desired optimization problem, the agent is tested in an OOTD environment and classifies said environment to one of its prior two belief clusters according to some distance function. Following the creation of the agent's belief cluster (cluster containing test environment), we then conduct the same training procedure again on this new environment with its belief cluster environments. If the new clustering result will separate the test environment in its own cluster, while others remain in the other one, we say the agent made a detection of the unknown causal factor. We run such experiments 10 times over different random seeds on different training-test environment pairs covering various unknown causal factors. To prevent over-fitting on In-Distribution tasks, training is performed on slightly different values of causal factors than what is seen during testing, e.g. train on $mass = 0.24m$, test on $mass = 0.20m$.

A.2 Modifying Environments

Mujoco. For mass experiments, we vary the normal mass of the robot $(m)$from $0.2m$ to $2m$. Similarly when modifying friction values in the environment, we change the friction coefficient η between the robot's actuators and the ground from 0.2η to 2η. For gravity experiments, we modify the absolute value. The ground truth ($g_z = -9.81$) from low gravity $g_z = -2.0$ to high gravity $g_z = -19.6$ for a total of 10 values. In wind experiments, we deviate from the typical value of 0.0 (no wind) for 10 values between 2.0 to 19.6. In **Causal World**, we are able to modify the absolute mass and shape of the block the agent interacts with. Changing the perception value of the robot is equivalent to modifying the skip-frame value of the robot's controller. Larger values of skip-frame leads to a slower refresh rate of the robot's sensors, and leads to less controllable actions.

B Probabilistic Baseline

To evaluate our prediction model, we design a baseline solely based on the first round of training. If the posterior distribution $\phi(s_t + 1|s_t, a_t)$ learned in the test environment is more than a reasonable large threshold distance away than the distribution learned in the training environments, as measured by KL divergence, we denote it as a detection.

We assume for a Causal POMDP $\mathbf{p}$, the agent's observations at timestep t is a random variable generated from a Gaussian distribution, such that $\mathbf{s}_{t+1} \sim \mathcal{N}(\mu_{s+t}, \Sigma)$. For each $(unseen, seen)$ pair of causal factors, we train an ensemble of probabilistic neural networks, each which output a mean vector $\mu_{\mathbf{p}}$ and a diagonal covariance matrix $\Sigma_{\mathbf{p}}$. Each ensemble is a uniformly-weighted mixture model, and we combine the predictions as $p(y|\mathbf{x}) = M^{-1}\sum_{m=1}^{M} p_{\theta_m}(y|\mathbf{x}, \theta_m)$. The prediction is then a mixture of Gaussian distributions. We assume the covariance matrix $\Sigma_{\mathbf{p}}$ is a diagonal matrix. For ease of computation, we further approximate the ensemble prediction as a Gaussian whose mean and variance are respectively that of the mixture; $\mu^* = M^{-1}\sum_{m=1}^{M}\mu_m$ and $\sigma^* = M^{-1}\sum_m(\sigma_m^2 + \mu_m^2) - \mu^{*2}$.

As training data for each network, we use the set of all $(state, action)$ pairs gathered during the first round of training our algorithm; $\mathcal{D} = \{(\mathbf{s}_t, \mathbf{a}_t), \mathbf{s}_{t+1}\}_{t=0}^{T}$.

In practice, each network was trained for 40 epochs across 10 random weight initializations, with a learning rate of 0.001 and Adam as the optimizer. We used an ensemble size of $M = 10$ for each experiment. To set the threshold, we gathered training data from 5 seeds unseen by our method, for every $(unseen, seen)$ pair of causal factors, and took the average value of the KL divergence of the test environment with respect to the training environments.

The ensemble model was inspired in part due to the observation that different random weight initialization produced different distribution predictions from one another. However, ultimately we remark that the over-

all performance of the baseline did not differ significantly if an ensemble was not used.

C Error Analysis

As is evident our result figures, agents are more likely to detect an unknown causal factor when a larger change is made to its value (larger values further away from the in-distribution value column). Agents are less likely to detect a change to their environment when the percentage change of the training causal factor in its belief cluster is large while the percentage change of the unknown causal factor is small. In Causal World, we found different factors to have different significance levels. In general, $Framerate >> Size > Damping > Mass >> Friction$. In an environment setting, the agent is able to detect a causal factor if the training factor has a lower significance value than the causal factor. For example, after examining the visualizations, we find that when the test environment is clustered together with heavy masses, the heavy mass dominates the effect of the damping, and the agent learns to further separate heavy blocks from light blocks in this new setting.

In another word, maximizing Causal Curiosity will separate the most significant factor (the significance is determined by the nature of the factor and the variance of it across all training environments) into 2 clusters. Each cluster will have a smaller variance of the training factor, thus lower significance. When we continue this process until the significance of the training factor is low enough in a cluster, the next significant factor (causal factor in our case) will be taken into consideration in the next training.

In Mujoco, after examining the visualizations, we postulate that agents with high action and observation spaces, such as Walker, are more prone to confusing actions such as front-flips and rolls with being pushed by the wind. This could be due to frequent relative change in position from one of the robot's sensors to another. Agents with small action and observation spaces, such as the Hopper, suffer less from this sensor confusion because their observations rely more on their absolute position in the environment. In Mujoco, a robot's absolute position in their environment was one of the most important factors in determining whether an environment OOTD or not for many of the considered causal factors.

Finally, compared to the probabilistic baseline, we would like to point out our method shows a more anthropomorphic response to varying values of causal factors. Consider the following example of how a human might see if its windy outside before leaving the house. A human may still need to check the weather report, or look at the leaves blowing in the wind, to determine if there is slight or no breeze outside. However, if there is a significant gust, one would simply be able to tell by sticking her arm out the window. Similarly, our method shows a similar (lack-of) sensitivity to certain varying causal factors. On the other hand, the baselines do not show such sensitivity. The tendency to predict the same value across multiple $(unseen, seen)$ pairs is likely due to the data generative process used to gather the training data. Not all action sequences may bring forth the causal factor's influence in the environment, but we consider all action sequences generated by the planner during the initial training process. Our method on the other hand, only considers the best action sequence; the action sequence which maximizes the optimization problem discussed in the main text.