Instructions for Paper Submissions to AISTATS 2022: Supplementary Materials

1 Implementation details

1.1 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 $mass = 0.20m$.

1.2 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.

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

3 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.