Fighting Spurious Correlations in Behavior Cloning with Multiple Environments

Han Lin
hl3199@columbia.edu
MS in Computer Science, Columbia University

Abstract

Offline imitation learning aims at using collected expert demonstrations to train learning agents. Traditional Behavioral Cloning (BC) algorithm in imitation learning directly maps input observations to expert actions, which is prone to spurious correlations (sometimes also called as causal confusions) due to the difference of observed variables from each training environment, and may not be able to generalize well. Therefore, it is tempting to ask could we design a method that explicitly learns the underlying causal structure to tackle this issue. However, recent literature [12] shows a sober look that unsupervised learning disentangled representations is fundamentally impossible without additional information on the models or data. So we need at least some additional information (e.g. multiple environments) to make the causal graph identifiable. Greatly inspired by the Nonlinear IRM model proposed by Lu et al. [13], we consider the setting of learning from expert's demonstrations from multiple environments, with the aim of generalizing well to new unseen environments. We make several adjustments of the original iCaRL three-phase procedure to adapt it to our imitation learning tasks, and proposed our new algorithm, Invariant Behavioral Cloning (IBC). We compare our method against several benchmarks on three OpenAI Gym control tasks and show its effectiveness in learning imitation policies capable of generalizing to new environments. Finally, to boost our understanding, we also conduct extensive ablation tests over different part of our algorithm, which we believe could inspire the direction of future research in causal imitation learning.

1 INTRODUCTION

Offline Imitation Learning (OIL) aims to learn a policy directly from existing expert demonstrations, which effectively avoids the need for costly environment interactions [16, 2]. For example, it would be dangerous and impractical for an agent to learn how to self-drive by making trails on real roads. The simplest method in OIL is Behavioral Cloning (BC), which solves a supervised learning problem over state-action pairs from expert demonstrations. However, recent findings suggest that BC suffers from spurious correlations (sometimes named as causal confusion), where the learned policy depends on some nuisance variables strongly correlated with expert actions or training environments, instead of the true causes. [8, 20, 7]. Fig. 1 in [8] provides a good illustration example for spurious correlations in self-driving where more information yields worse imitation learning performance.

Illustrative Example: Consider using imitation learning to train a learner to drive a car (see Fig. 1)[8]. In scenario A, the model's input is an image of the dashboard and windshield, where the dashboard has a *indicator light* that comes on immediately when the brake is applied. While in scenario B, the input to the model (with identical architecture) is the same image but with the dashboard masked out. Both cloned policies achieve low training loss, but when tested on the road, model B drives well, while model A does not. This is because model A wrongly learns to apply the brake only when the brake light is on. Even though the brake light is the **effect** of braking, model A could achieve low training error by misidentifying it as the **cause** instead.

Preprint. Under review.

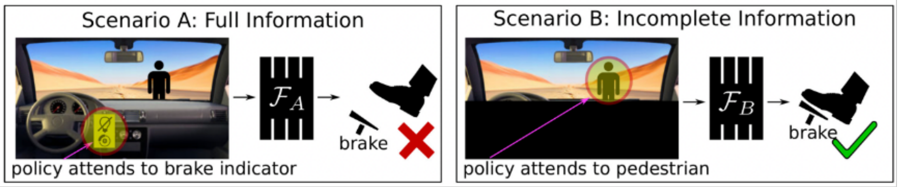
Scenario A: Full Information
$\mathcal{F}_A$
brake
policy attends to brake indicator
Scenario B: Incomplete Information
$\mathcal{F}_B$
brake
policy attends to pedestrian

Figure 1: Example of spurious correlation (causal confusion). Directly applying Behavioral Cloning (Scenario A) will make the agents incorrectly learn the indicator light as cause of brake.

If we rethink about this example, we would notice that the *indicator light* is perfectly correlated with the brakes. Therefore, it would be impossible to distinguish whether such *indicator light* is the cause or effect of taking brakes. [8] provides a smart but simple idea to eliminate such nuisance variables: rather than learning the underlying causal structure directly, it randomly mask out a certain percentage of latent variables (e.g. features learned from VAEs on image data) during the training process. If the training performance does not deteriorate for a certain mask, then the variables masked out are unlikely to be the direct parent of expert actions, and thus could be removed. Spurious correlations can also be eliminated in this way since the expert's policy is independent on these variables. Such method works well in several OIL tasks including OpenAI Gym and Atari Games [8, 14].

However, we are still curious about whether there exist better methods that could identify these nuisance variables by learning the causal structure directly. Sadly, recent progress shows a sober look that unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the data [12]. On the other hand, if we have expert demonstrations from multiple environments, it would be possible to identify the causal structure under certain assumptions [13]. Methods including Invariant Risk Minimization (IRM) is widely used when learn from multiple environments [3].

Our Contributions: With motivations stated above, we study offline imitation learning with expert demonstrations from multiple environments in this paper. Our contribution is three folds:

- To the best of our knowledge, we are the first open source implementation of the NF-iVAE model proposed by Lu et al. [13], and the first to apply it to tasks in imitation learning.
- We adapt several parts in iCaRL [13] to make it work for our case.
- We make extensive ablation tests to show the effectiveness of each part of our proposed algorithm.

2 RELATED WORK

Causal confusion problem happens when a policy exploits the nuisance correlated in states for predicting expert actions. Distribution shift is one of the main reason that cause such confusion. Therefore, various research focus on learning disentangled representations which is robust to distribution shift. Haan [8] solves the causal confusion problem by randomly masked disentangled representations learned from β-VAE, and infers the best mask through environment interaction. Instead, [14] removes the requirement of environment interaction, and achieves better performance by learning disentangled representations through VQ-VAE. Some reviewers [1] doubt that the resolution of the Atari datasets used for experiments in [14] is not high enough, recent models including VQ-VAE2 [15] and VQ-GAN [9], which focus on large images, might be a promising way to solve their concerns. This line of work mainly focus on learning disentangled representations, and use **random masking** to sample the best subset of representation variables in the hope of removing nuisance correlates.

Considering the above methods do not learn causal graphs explicitly, we may wonder are there ways to solve causal confusion by directly learning causal graphs. However, recent progress shows a sober look that unsupervised learning of disentangled representations is fundamentally impossible without inductive biases on both the models and the data [12]. In other words, the causal confusion problem cannot be solved completely by just collecting more samples from expert, as there is no additional information for identifying the cause of expert actions, and the reward function is just learned using the given expert demonstrations where nuisance correlates exist [1]. Therefore, recent literatures for learning disentangled representations focus on using some additional labeled data to help identify the causal relationship [21, 18], but it's still unclear how can we apply these methods to imitation learning.

To the best of our knowledge, there's **no former research in imitation learning that explicitly solves causal confusion problem by learning the causal graphs directly** due to the obstacles stated above. Combining causal models and representation learning in imitation learning still remains an open problem [17].

Imitation Learning

Spurious Correlations & Causal Confusion & Copycat

Invariant Risk Minimization:

Variational Antoencoders (VAEs):

3 PRELIMINARIES

3.1 Notations

The notations commonly used in imitation learning (IL) is a little bit different from causal inference (CI), and we try to make a compromise between these two. So let's first be clear about some definitions we'll use later in this paper:

$\mathbf{X}$: observations (usually denoted as **O** or $\mathbf{X}$ in causal inference, and **S** in imitation learning).

A: actions (equivalent to label **Y** in causal inference).

π: imitation policy, which is a function (e.g. neural network) that maps from $\mathbf{X}$ to $\mathbf{A}$.

Z: latent states (borrowed from causal inference, not often used in imitation learning).

E: environments.

PA$(\mathbf{A})$: direct causes (parents) of $\mathbf{A}$.

3.2 Problem Formulation

We introduce the standard imitation learning framework with multiple environments in this section. Consider $\mathcal{M} = \{(\mathcal{X}^e, \mathcal{A}, \mathcal{P}^e, r^e, \gamma) | e \in \mathcal{E}\}$ as a set of environments with observations $x^e \in \mathcal{X}^e$, actions $a \in A$, transition probabilities $\mathcal{P}^e : \mathcal{X} \times \mathcal{A} \rightarrow \mathcal{X}$, rewards $r^e \in \mathbb{R}^{\mathcal{X} \times \mathcal{A}}$, and discount factor γ. $\mathcal{X}, \mathcal{P}$ and r^e are environment-dependent, while $\mathcal{A}$ and γ are shared across all environments. We have access to an offline training dataset consists of expert demonstrations from multiple environments $\mathcal{D} = \{\{(x_i, a_i)\}_{i=1}^{N^e} | e \in \mathcal{E}_{\text{train}}\}$, with each data point contains an observation x_i together with the expert's corresponding action a_i after observing x_i. Such offline setting, which restricts the learning agent from interacting with the environments, is very common in practice when interacting with environment is very expensive or impossible (e.g. teaching a car to self-drive by learning from expert demonstrations).

Our goal is to learn a policy $\pi : \mathcal{X} \rightarrow \mathcal{A}$ that matches the expert's policy $\pi_{\mathcal{D}}$ across all environments, and even generalizes well to some new environments. If we define $\mathcal{L}^e$ as the imitation loss (cross-entropy loss), then we could formalize our goal as finding a policy π that minimizes the maximum imitation loss across all environments:

$$\min_{\pi} \max_{e \in \mathcal{E}} \mathcal{L}^e(\pi, \pi_{\mathcal{D}}) \tag{1}$$

4 Methods

In this section, we first introduce several closely related algorithms, then propose our algorithms.

4.1 Comparison Benchmarks

Behavioral Cloning (BC): The most commonly seen method for imitation learning is behavioral cloning, which uses supervised learning techniques for imitation learning. To be more specific, let's denote f as an encoder that maps observations x_t to a low-dimensional space. Behavioral Cloning learns a policy π that maps the feature map $f(x_t)$ to the expert action a_t. π and f are learned by minimizing the negative log-likelihood of expert actions as follows:

$$\mathcal{L}_{\text{BC}}(x_t, a_t) = -\log \pi(a_t | f(x_t)) \tag{2}$$

Behavioral Cloning + Invariant Risk Minimization (BC+IRMv1): We borrow the idea of Invariant Risk Minimization (IRM) [3] into our imitation learning settings. Since our classifier (policy function) is non-linear, we use the IRM-v1 objective which has a penalty term instead of hard constraints. In our case, $R^e = R^e_{\text{BC}} =$ Cross Entropy$(\pi(.|f(x_t)), a_t)$, and we are aiming to find a policy π that is invariant across different environments.

$$\mathcal{L}_{\text{IRM}}(\Phi, \omega) = \sum_{e \in \mathcal{E}_{\text{train}}} R^e(\bar{\omega} \circ \Phi) + \lambda \|\nabla_{\omega|\omega=1.0} R^e(\bar{\omega} \circ \Phi)\| \tag{3}$$

CCIL (Haan et al. [8]): This method first use β-VAE to learn an disentangled representation $f(X) \in \mathbb{R}^d$ from images. Since each dimension of this disentangled representation could either be a direct cause of expert action (PA(A)) or not, there are 2^n possible graphs. Then the author parameterize the structure G of the causal graph as a vector of n binary variables, each indicating the presence of an arrow from $f(X_i)$ to A. In order to learn the causal graph, G is drawn uniformly at random over all 2^n possible graphs, and the paper minimizes the following loss, where $\odot$ is elementwise product, and $[,]$ represents concatenation. $\pi : f(X) \to A$ is the policy network to be trained. After training, the graph G^* that minimizes the loss $\mathcal{L}_{\text{CE}}$ inside expectation will be the best approximation for the underlying causal graph.

$$\mathbb{E}_G[\mathcal{L}_{\text{CE}}(\pi([f(X_i) \odot G, G]), A)] \tag{4}$$

4.2 Our Methods: Invariant Behavioral Cloning (IBC)

Our proposed algorithm, Invariant Behavioral Cloning, is adapted from iCaRL (Lu et al. [13]). Analogy to their paper, we also divide our approach into three phases as shown in Algorithm 1.

Algorithm 1: Invariant Behavioral Cloning (IBC)

```
$mark[u] = 1$
Phase1: First learn a NF-iVAE model by optimizing the objective function in Eq. (??) on the
  data $\mathcal{D} = \{\{(x_i, a_i)\}_{i=1}^{N^e} | e \in \mathcal{E}_{\text{train}}\}$. Then we use the mean of the NF-iVAE encoder to infer
  the latent variables $\mathbf{Z}$ from observations.
Phase2: After inferring $\mathbf{Z}$, we can discover direct causes (parents) of $\mathbf{A}$ by testing all pairs of
  latent variables with Fast Conditional Independence Test (FCIT), i.e. finding a set of latent
  variables in which each pair of $Z_i$ and $Z_j$ satisfies that the dependency between them increases
  after conditioning on $A$.
Phase3: Having obtained PA$(\mathbf{Y})$, we can solve the optimization problems similar to the IRM
  objective function in Eq. (8). Different from [13], we learn another encoder that maps from
  observations $\mathbf{X}$ to PA$(\mathbf{Y})$. When in a new environment, we use this encoder to infer PA$(\mathbf{Y})$
  from observations, and then leverage the learned policy $\pi$ for action prediction.
```

4.2.1 Phase 1: Identifying Latent Variables

In phase 1, we replicate the NF-iVAE model in [13], which is an extended iVAE with non-factorized prior that is able to capture complex structures in the latent states $\mathbf{Z}$. To be more specific, we are aiming at jointly learn ($\mathbf{f}$, $\mathbf{T}$, λ, ϕ) from the following objective:

$$\mathcal{L}_{\text{phase1}}(\theta, \phi) = \mathcal{L}^{\text{VAE}}_{\text{phase1}}(f, \hat{T}, \hat{\lambda}, \phi) - \mathcal{L}^{\text{SM}}_{\text{phase1}}(\hat{f}, T, \lambda, \hat{\phi}) \tag{5}$$

where we use $\hat{f}, \hat{T}, \hat{\lambda}, \hat{\phi}$ to represent copies of f, t, λ, ϕ that are treated as constants and whose gradient is not calculated during training. $\mathcal{L}^{\text{VAE}}_{\text{phase1}}$ and $\mathcal{L}^{\text{SM}}_{\text{phase1}}$ on the right hand side are:

$$\mathcal{L}^{\text{VAE}}_{\text{phase1}}(f, \hat{T}, \hat{\lambda}, \phi) = \mathbb{E}_{p_D}[\mathbb{E}_{q_\phi(Z|X,A,E)}[\log p_f(X|Z) + \log p_{\hat{T},\hat{\lambda}}(Z|A,E) - \log q_\phi(Z|X,A,E)]] \tag{6}$$

$$\mathcal{L}^{\text{SM}}_{\text{phase1}}(\hat{f}, T, \lambda, \hat{\phi}) = \mathbb{E}_{p_D}[\mathbb{E}_{q_{\hat{\phi}}(Z|X,A,E)}[\|\nabla_Z \log q_{\hat{\phi}}(Z|X,A,E) - \nabla_Z \log p_{T,\lambda}(Z|A,E)\|^2]] \tag{7}$$

we call $q_\phi(Z|X,A,E)$ as encoder, $p_f(X|Z)$ as decoder, and $p_{T,\lambda}(Z|A,E)$ as non-factorized prior.

ablation test over other VAEs are in the appendix

4.2.2 Phase 2: Discovering Direct Causes

With the encoder which maps from observations $\mathbf{X}$ to latent states $\mathbf{Z}$ we estimated in phase 1, we could now discover direct causes (parents) of actions $\mathbf{A}$. [13] uses independence testing [11] and conditional independence testing [22] in their paper. However, the imitation learning tasks we tested in the Sec. (5) have discrete actions $\mathbf{A}$, while HSIC and KCIT work for continuous variables. Instead, we choose to use Fast (Conditional) Independence Test (FIT and FCIT)[1] [6], which is a quite flexible method that works for both continuous and discrete/categorical variables.

With such method, we could test all pairs of latent variables $\mathbf{Z}$ by comparing p-value from the independence test (FIT) and conditional independence test (FCIT). [13] does not told explicitly the precedure If the p-value from conditional independence test is larger than the p-value from independence test for $k \geq \lceil d/2 - 1 \rceil$[2] times among the pairs including latent state $\mathbf{Z_i}$, then we regard Z_i as one of the direct parents of actions $\mathbf{A}$. Ablation tests over k is given in the appendix.

4.2.3 Phase 3: Learning an Invariant Policy

With direct parents of actions $\mathbf{A}$, we can learn an invariant policy π by solving the following optimization problem:

$$\min_\pi \sum_{e \in \mathcal{E}_{\text{train}}} R^e(\pi) = \min_\pi \sum_{e \in \mathcal{E}_{\text{train}}} \mathbb{E}_{\text{PA}(\mathbf{A}^e),\mathbf{A}^e}[\mathcal{L}_{\text{CE}}(\pi(\text{PA}(\mathbf{A}^e)), \mathbf{A}^e)] \tag{8}$$

which complete the whole training steps.

Then in a new testing environment, we need to infer PA$(\mathbf{A})$ from observations $\mathbf{X}$. [13, 19] optimize the following objective to infer PA$(\mathbf{A})$ from X.

$$\max_{\mathbf{Z}} \log p_f(X|\text{PA}(\mathbf{A}), \neg\text{PA}(\mathbf{A})) + \lambda_1 \|\text{PA}(\mathbf{A})\|_2^2 + \lambda_2 \|\neg\text{PA}(\mathbf{A})\|_2^2 \tag{9}$$

However, such method suffers two shortcomings we observed during our experiments:

(1) λ_1, λ_2 are hard to tune in practice. Sub-optimal values of these two parameters might change the scale of PA$(\mathbf{A})$ and $\neg$PA$(\mathbf{A})$, which might drift from the scale of actual latent states $\mathbf{Z}$.

(2) we need to run such optimization procedure for each evaluated observation in $\mathbf{X}$. This is not very realistic in our imitation learning tasks which usually contains 100 evaluation episodes, with each episode contains 500-1000 observations.

Given a new observation $\mathbf{X}_{\text{new}}$, another way we have considered is to choose the latent states in our training data whose corresponding observation $\mathbf{X}$ is closest to $\mathbf{X}_{\text{new}}$. However, this method also can't work well in practice. Therefore, we use a more brute force approach to learn a new encoder (neural networks) that maps observations $\mathbf{X}$ to latent states $\mathbf{Z}$. ablation tests over these three methods are given in the appendix

[1] `https://github.com/kjchalup/fcit`

[2] we use d to represent the dimension of latent space

5 Experiments

In this section, we perform experiments on both image and non-image data related to imitation learning to test the effectiveness of different methods. For image data, we use the Pong Game[1] from Atari environment. For non-image data, we tested on three commonly used imitation learning control tasks from OpenAi Gym [5]: Acrobot [10], Cartpole [4] and LunarLander [5].

5.1 OpenAi Gym Tasks

Data Preparation: For each task, we use DQN to train the expert policies. Then we generate 2000 trajectories of expert demonstrations, with each trajectory contains up to 500 state-action pairs. To incorporate spurious correlations into our dataset, we concatenate four spurious correlated states to the oroginal state space. The first three spurious states are generated from linear combinations of the original states with different multiplicative factors ($1\times$ and $2\times$ for the first and second environment respectively), and we use an environment identifier as the last spurious state. Details for such data-generating process is contained in the appendix.

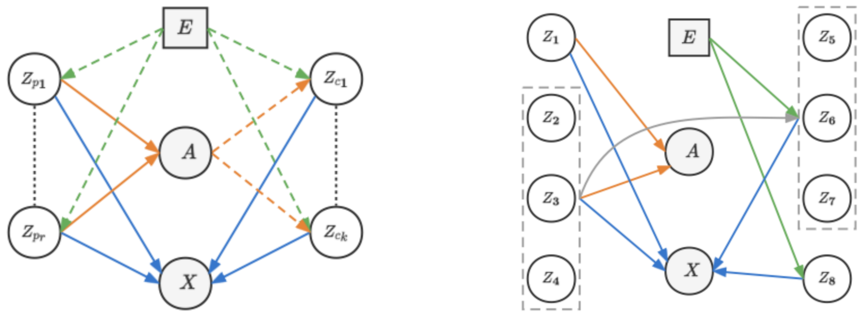

Figure 2: Left: The general causal graph in [13]. Grey dashed lines represent arbitrary connections between the latent variables, Dashed green and orange arrow lines denote the edges which might vary across environments and might be abasent. Right: Causal graph for our CartPole task. To keep clean of the causal graph, the latent variables in the dashed rectangles have the same in/out edges to other variables. And the grey solid line between two rectangles means that Z_5 to Z_7 (spurious correlated latent states) are generated from linear combinations of the original latent states Z_2 to Z_4. Z_5 to Z_7 are also influenced by the environment.

Table 1: OpenAI Gym Datasets Descriptions.

	Observation Space	**Action Space**	**Random Score**	**Expert Score**
Acrobot-v1	6 (Continuous)	3 (Discrete)	-439.92 ± 13.14	-87.32 ± 12.02
CartPole-v1	4 (Continuous)	2 (Discrete)	19.12 ± 1.76	500.00 ± 0.00
LunarLander-v2	8 (Continuous)	4 (Discrete)	-452.22 ± 61.24	271.71 ± 17.88

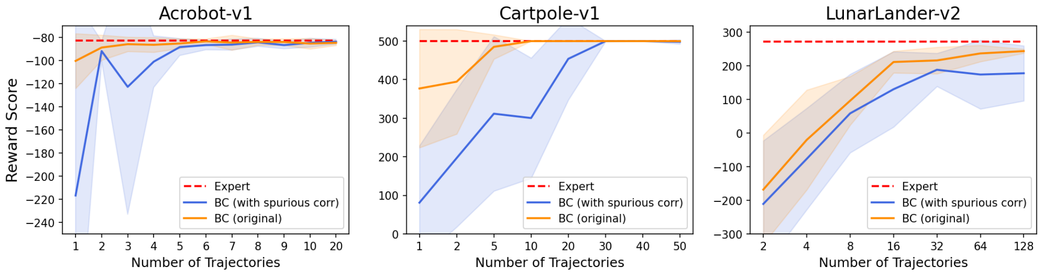

Figure 3: Reward (y-axis) vs number of training trajectories from expert demonstrations (x-axis) for dataset before (orange line) and after (blue line) adding spurious correlations. Expert scores are marked as red dashed lines. The shaded areas represent 0.5*standard deviation. Results are averaged over 15 runs.

[1] https://www.endtoend.ai/envs/gym/atari/pong/

Figure 5.1 shows the rewards of Behavioral Cloning strategy under (1) original state space and (2) augmented state space with spurious correlations. The conclusion is clear: across these tasks, learning a policy from environment-dependent states leads to inferior performance.

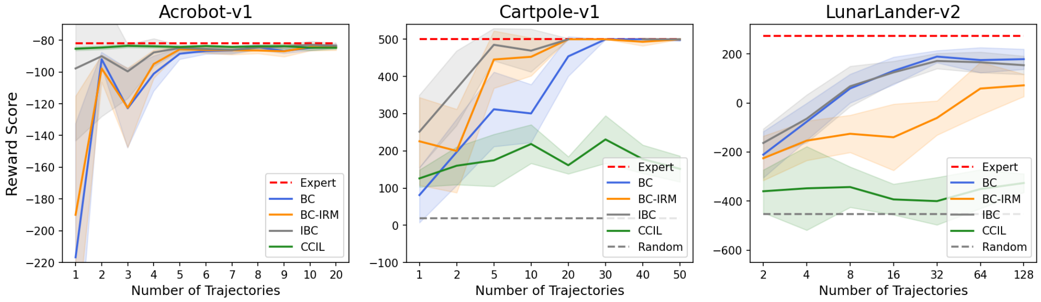

Figure 4: Reward (y-axis) vs number of training trajectories from expert demonstrations (x-axis) for different methods. Expert scores and random scores are marked as red and grey dashed lines respectively. The shaded areas represent 0.25*standard deviation. Results are averaged over 15 runs.

6 ABLATION TESTS

6.1 Phase 1 Ablation: Different Assumption on the Prior

:

6.2 Phase 2 Ablation: Parent Selection Threshold

:

6.3 Phase 3 Ablation: Infer $\mathrm{PA}(A)$ from X in A New Environment

:

References

[1] Object-aware regularization for addressing causal confusion in imitation learning, author's response for reviewer jmzw. `https://openreview.net/forum?id=FEhntTXAeHN`.

[2] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[3] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization, 2019.

[4] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983.

[5] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

[6] K. Chalupka, P. Perona, and F. Eberhardt. Fast conditional independence test for vector variables with large sample sizes, 2018.

[7] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. *CoRR*, abs/1904.08980, 2019.

[8] P. de Haan, D. Jayaraman, and S. Levine. Causal confusion in imitation learning. *CoRR*, abs/1905.11979, 2019.

[9] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. *CoRR*, abs/2012.09841, 2020.

[10] A. Geramifard, C. Dann, R. H. Klein, W. Dabney, and J. P. How. Rlpy: A value-function-based reinforcement learning framework for education and research. *Journal of Machine Learning Research*, 16(46):1573–1578, 2015.

[11] A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems*, volume 20. Curran Associates, Inc., 2007.

[12] F. Locatello, S. Bauer, M. Lucic, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *CoRR*, abs/1811.12359, 2018.

[13] C. Lu, Y. Wu, J. M. Hernández-Lobato, and B. Schölkopf. Nonlinear invariant risk minimization: A causal approach. *CoRR*, abs/2102.12353, 2021.

[14] J. Park, Y. Seo, C. Liu, L. Zhao, T. Qin, J. Shin, and T. Liu. Object-aware regularization for addressing causal confusion in imitation learning. *CoRR*, abs/2110.14118, 2021.

[15] A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. *CoRR*, abs/1906.00446, 2019.

[16] S. Schaal. Learning from demonstration. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9. MIT Press, 1996.

[17] B. Schölkopf. Causality for machine learning. *CoRR*, abs/1911.10500, 2019.

[18] X. Shen, F. Liu, H. Dong, Q. LIAN, Z. Chen, and T. Zhang. Disentangled generative causal representation learning, 2021.

[19] X. Sun, B. Wu, C. Liu, X. Zheng, W. Chen, T. Qin, and T. Liu. Latent causal invariant model. *CoRR*, abs/2011.02203, 2020.

[20] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao. Fighting copycat agents in behavioral cloning from observation histories. In *NeurIPS*, 2020.

[21] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang. Causalvae: Structured causal disentanglement in variational autoencoder. *CoRR*, abs/2004.08697, 2020.

[22] K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. *CoRR*, abs/1202.3775, 2012.

7 APPENDIX

7.1 Atari Pong Games Task

Data Description: Pong from the Atari environment is a two-dimensional sports game that simulates table tennis. A player can control the right green paddle to hit a ball back. And the goal for each player is to reach 11 points before the opponent. The imitator has access to discrete video images x_t as well as expert's action a_t for a total of T periods. And the imitator's goal is to learn a policy π from $\{(x_t, a_t)\}_{t=1}^{T}$ to achieve higher score of the game.

Environments Generation: In order to create environments with spurious correlations, we use (a) the original image with scores of agents at the top and (b) image augmented with a number representing previous action at the left bottom corner as two different training environments. We evaluate the trained policy on a new environment (c), which contains neither scores nor previous actions. The scores and actions in (a) and (b) are nuisance variables independent with the expert's actions.

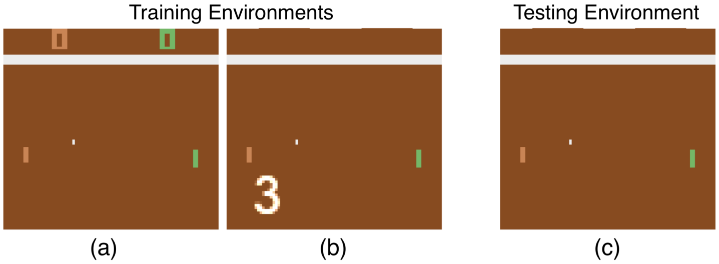

Figure 5: Two training environments (a) and (b) and one testing environment (c) on Atari Pong Games. (a) has scores of each agent at the top of the image, (b) has expert's previous action as a number at the left corner of the image, and (c) has neither scores nor previous action showing up in the image.