
Causal Confusion in Imitation Learning

Pim de Haan Dinesh Jayaraman Sergey Levine
Berkeley Artificial Intelligence Research, UC Berkeley

Abstract

Behavioral cloning reduces policy learning to supervised learning by training a discriminative model to predict expert actions given observations. Such discriminative models are non-causal: the training procedure is unaware of the causal structure of the interaction between the expert and the environment. We point out that ignoring causality is particularly damaging because of the distributional shift in imitation learning. In particular, it leads to a counter-intuitive “causal confusion” phenomenon: access to more information can yield worse performance. We investigate how this problem arises, and propose a solution to combat it through targeted interventions—either environment interaction or expert queries—to determine the correct causal model. We show that causal confusion occurs in several benchmark control domains as well as realistic driving settings, and validate our solution against DAGger and other baselines and ablations.

1 Introduction

Imitation learning allows for control policies to be learned directly from example demonstrations provided by human experts. It is easy to implement, and reduces or removes the need for extensive interaction with the environment during training [56, 39, 4, 1, 18].

However, imitation learning suffers from a fundamental problem: distributional shift [8, 40]. Training and testing state distributions are different, induced respectively by the expert and learned policies. Therefore, imitating expert actions on expert trajectories may not align with the true task objective. While this problem is widely acknowledged [39, 8, 40, 41], yet with careful engineering, naïve behavioral cloning approaches have yielded good results for several practical problems [56, 39, 42, 34, 35, 4, 31, 3]. This raises the question: is distributional shift really still a problem?

In this paper, we identify a somewhat surprising and very problematic effect of distributional shift: “causal confusion.” Distinguishing correlates of expert actions in the demonstration set from true causes is usually very difficult, but may be ignored without adverse effects when training and testing distributions are identical (as assumed in supervised learning), since nuisance correlates continue to hold in the test set. However, this can cause catastrophic problems in imitation learning due to distributional shift. This is exacerbated by the causal structure of sequential action: the very fact that current actions cause future observations often introduces complex new nuisance correlates.

To illustrate, consider behavioral cloning to train a neural network to drive a car. In scenario A, the model’s input is an image of the dashboard and windshield, and in scenario B, the input to the model (with identical architecture) is the same image but with the dashboard masked out (see Fig 1). Both cloned policies achieve low training loss, but when tested on the road, model B drives well, while model A does not. The reason: the dashboard has an indicator light that comes on immediately when the brake is applied, and 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.

This situation presents a give-away symptom of causal confusion: access to *more information* leads to *worse generalization performance* in the presence of distributional shift. Causal confusion occurs

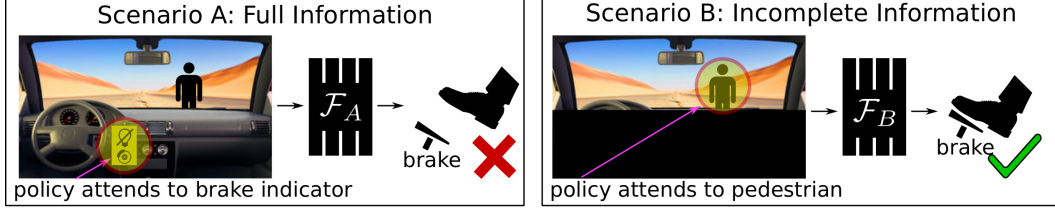


Figure 1: Causal confusion: *more* information yields worse imitation learning performance. Model A relies on the braking indicator to decide whether to brake. Model B instead correctly attends to the pedestrian.

commonly in natural imitation learning settings, especially when the imitator’s inputs include history information.

In this paper, we first point out and investigate the causal confusion problem in imitation learning. Then, we propose a solution to overcome it by learning the correct causal model, even when using complex deep neural network policies. We learn a mapping from causal graphs to policies, and then use targeted interventions to efficiently search for the correct policy, either by querying an expert, or by executing selected policies in the environment.

2 Related Work

Imitation learning. Imitation learning through behavioral cloning dates back to Widrow and Smith, 1964 [56], and has remained popular through today [39, 42, 34, 35, 4, 11, 31, 54, 3]. The distributional shift problem, wherein a cloned policy encounters unfamiliar states during autonomous execution, has been identified as an issue in imitation learning [39, 8, 40, 41, 23, 17, 3]. This is closely tied to the “feedback” problem in general machine learning systems that have direct or indirect access to their own past states [45, 2]. For imitation learning, various solutions to this problem have been proposed [8, 40, 41] that rely on iteratively querying an expert based on states encountered by some intermediate cloned policy, to overcome distributional shift; DAgger [41] has come to be the most widely used of these solutions.

We show evidence that the distributional shift problem in imitation learning is often due to causal confusion, as illustrated schematically in Fig 1. We propose to address this through targeted interventions on the states to learn the true causal model to overcome distributional shift. As we will show, these interventions can take the form of either environmental rewards with no additional expert involvement, or of expert queries in cases where the expert is available for additional inputs. In expert query mode, our approach may be directly compared to DAgger [41]: indeed, we show that we successfully resolve causal confusion using orders of magnitude fewer queries than DAgger.

We also compare against Bansal et al. [3]: to prevent imitators from copying past actions, they train with dropout [51] on dimensions that might reveal past actions. While our approach seeks to find the true causal graph in a mixture of graph-parameterized policies, dropout corresponds to directly applying the mixture policy. In our experiments, our approach performs significantly better.

Causal inference. Causal inference is the general problem of deducing cause-effect relationships among variables [50, 36, 38, 48, 9, 49]. “Causal discovery” approaches allow causal inference from pre-recorded observations under constraints [52, 15, 27, 13, 28, 29, 24, 12, 32, 55]. Observational causal inference is known to be impossible in general [36, 37]. We operate in the interventional regime [53, 10, 47, 46] where a user may “experiment” to discover causal structures by assigning values to some subset of the variables of interest and observing the effects on the rest of the system. We propose a new interventional causal inference approach suited to imitation learning. While ignoring causal structure is particularly problematic in imitation learning, ours is the first effort directly addressing this, to our knowledge.

3 The Phenomenon of Causal Confusion

In imitation learning, an expert demonstrates how to perform a task (e.g., driving a car) for the benefit of an agent. In each demo, the agent has access both to its n -dim. state observations at each time t , $X^t = [X_1^t, X_2^t, \dots, X_n^t]$ (e.g., a video feed from a camera), and to the expert’s action A^t (e.g., steering,

acceleration, braking). Behavioral cloning approaches learn a mapping π from X^t to A^t using all (X^t, A^t) tuples from the demonstrations. At test time, the agent observes X^t and executes $\pi(X^t)$.

The underlying sequential decision process has complex causal structures, represented in Fig 2. States influence future expert actions, and are also themselves influenced by past actions and states.

In particular, expert actions A^t are influenced by *some* information in state X^t , and unaffected by the rest. For the moment, assume that the dimensions $X_1^t, X_2^t, X_3^t, \dots$ of X^t represent disentangled factors of variation. Then some unknown subset of these factors (“causes”) affect expert actions, and the rest do not (“nuisance variables”). A confounder $Z^t = [X^{t-1}, A^{t-1}]$ influences each state variable in X^t , so that some nuisance variables may still be correlated with A^t among (X^t, A^t) pairs from demonstrations. In Fig 1, the dashboard light graph represents a confounder.

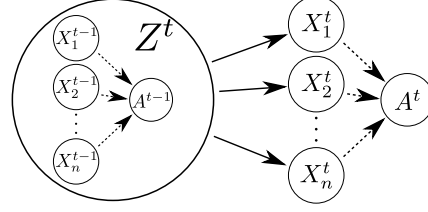


Figure 2: Causal dynamics of imitation. Parents of a node represent its causes.

A naïve behavioral cloned policy might rely on nuisance correlates to select actions, producing low training error, and even generalizing to held-out (X^t, A^t) pairs. However, this policy must contend with distributional shift when deployed: actions A_t are chosen by the *imitator* rather than the expert, affecting the distribution of Z_t and X^t . This in turn affects the policy mapping from X^t to A^t , leading to poor performance of expert-cloned policies. We define “causal confusion” as the phenomenon whereby cloned policies fail by misidentifying the causes of expert actions.

3.1 Robustness and Causality in Imitation Learning

Intuitively, distributional shift affects the relationship of the expert action A^t to nuisance variables, but not to the true causes. In other words, to be maximally robust to distributional shift, a policy must rely solely on the true causes of expert actions, thereby avoiding causal confusion. This intuition can be formalized in the language of functional causal models (FCM) and interventions [36].

Functional causal models: A functional causal model (FCM) over a set of variables $\{Y_i\}_{i=1}^n$ is a tuple (G, θ_G) containing a graph G over $\{Y_i\}_{i=1}^n$, and deterministic functions $f_i(\cdot; \theta_G)$ with parameters θ_G describing how the causes of each variable Y_i determine it: $Y_i = f_i(Y_{\text{Pa}(i; G)}, E_i; \theta_G)$, where E_i is a stochastic noise variable that represents all external influences on Y_i , and $\text{Pa}(i; G)$ denote the indices of parent nodes of Y_i , which correspond to its causes.

An “intervention” $do(Y_i)$ on Y_i to set its value may now be represented by a structural change in this graph to produce the “mutilated graph” $G_{\bar{Y}_i}$, in which incoming edges to Y_i are removed.¹

Applying this formalism to our imitation learning setting, any distributional shift in the state X^t may be modeled by intervening on X^t , so that correctly modeling the “interventional query” $p(A^t | do(X^t))$ is sufficient for robustness to distributional shifts. Now, we may formalize the intuition that only a policy relying solely on true causes can robustly model the mapping from states to optimal/expert actions under distributional shift.

In Appendix B, we prove that under mild assumptions, correctly modeling interventional queries does indeed require learning the correct causal graph G . In the car example, “setting” the brake light to on or off and observing the expert’s actions would yield a clear signal unobstructed by confounders: the brake light does not affect the expert’s braking behavior.

3.2 Causal Confusion in Policy Learning Benchmarks and Realistic Settings

Before discussing our solution, we first present several testbeds and real-world cases where causal confusion adversely influences imitation learning performance.

Control Benchmarks. We show that causal confusion is induced with small changes to widely studied benchmark control tasks, simply by adding more information to the state, which intuitively ought to make the tasks easier, not harder. In particular, we add information about the previous action, which tends to correlate with the current action in the expert data for many standard control problems. This is a proxy for scenarios like our car example, in which correlates of past actions are observable

¹For a more thorough overview of FCMs, see [36].

in the state, and is similar to what we might see from other sources of knowledge about the past, such as memory or recurrence. We study three kinds of tasks: (i) MountainCar (continuous states, discrete actions), (ii) MuJoCo Hopper (continuous states and actions), (iii) Atari games: Pong, Enduro and UpNDown (states: two stacked consecutive frames, discrete actions).

For each task, we study imitation learning in two scenarios. In scenario A (henceforth called "CONFOUNDED"), the policy sees the augmented observation vector, including the previous action. In the case of low-dimensional observations, the state vector is expanded to include the previous action at an index that is unknown to the learner. In the case of image observations, we overlay a symbol corresponding to the previous action at an unknown location on the image (see Fig 3). In scenario B ("ORIGINAL"), the previous action variable is replaced with random noise for low-dimensional observations. For image observations, the original images are left unchanged. Demonstrations are generated synthetically as described in Appendix A. In all cases, we use neural networks with identical architectures to represent the policies, and we train them on the same demonstrations.

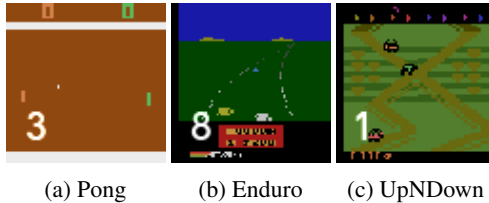


Figure 3: The Atari environments with indicator of past action (white number in lower left).

Fig 4 shows the rewards against varying demonstration dataset sizes for MountainCar, Hopper, and Pong. Appendix D shows additional results, including for Enduro and UpNDown. All policies are trained to near-zero validation error on held-out expert state-action tuples. ORIGINAL produces rewards tending towards expert performance as the size of the imitation dataset increases. CONFOUNDED either requires many more demonstrations to reach equivalent performance, or fails completely to do so.

Overall, the results are clear: across these tasks, access to *more* information leads to inferior performance. As Fig 10 in the appendix shows, this difference is not due to different training/validation losses on the expert demonstrations—for example, in Pong, CONFOUNDED produces lower validation loss than ORIGINAL on held-out demonstration samples, but produces lower rewards when actually used for control. These results not only validate the existence of causal confusion, but also provides us with testbeds for investigating a potential solution.

Real-World Driving. In more realistic imitation learning settings too, symptoms of causal confusion have been observed consistently in Muller et al. [34], Wang et al. [54], Bansal et al. [3], when learning to drive from histories of video frames. While these histories contain valuable information for driving, they also naturally introduce information about nuisance factors such as previous actions. In all three cases, more information led to worse results for the behavioral cloning policy, but this was neither attributed specifically to causal confusion, nor tackled using causally motivated approaches.

We draw the reader’s attention to particularly telling results from Wang et al. [54] for learning to drive in near-photorealistic GTA-V [22] environments, using behavior cloning with DAGger-inspired expert perturbation. Imitation learning policies are trained using overhead image observations with and without “history” information (HISTORY and NO-HISTORY) about the ego-position trajectory of the car in the past.

Metrics → Methods ↓	Validation Perplexity	Driving Performance		
		Distance	Interventions	Collisions
HISTORY	0.834	144.92	2.94 ± 1.79	6.49 ± 5.72
NO-HISTORY	0.989	268.95	1.30 ± 0.78	3.38 ± 2.55

Table 1: Imitation learning results from Wang et al. [54]. Accessing history yields better validation performance, but worse actual driving performance.

Similar to our tests above, architectures are identical for the two methods. And once again, like in our tests above, HISTORY has better performance on held-out demonstration data, but much worse performance when actually deployed. Tab 1 shows these results, reproduced from Wang et al. [54] Table II. These results constitute strong evidence for the prevalence of causal confusion in realistic imitation learning settings. Bansal et al. [3] also observe similar symptoms in a driving setting, and present a dropout [51] approach to tackle it, which we compare to in our experiments.

4 Resolving Causal Confusion

Recall from Sec 3.1 that robustness to causal confusion can be achieved by finding the true causal model of the expert’s actions. We propose a simple pipeline to do this. First, we jointly learn

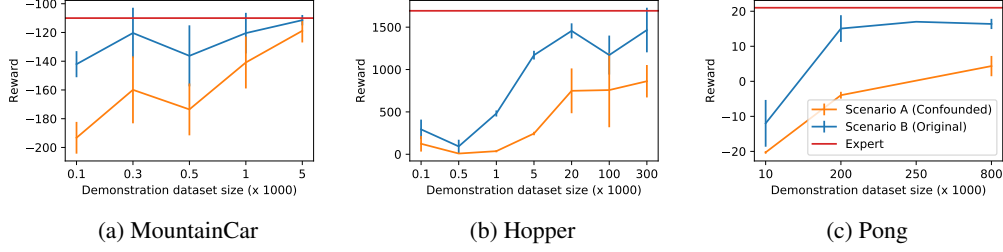


Figure 4: Diagnosing causal confusion: net reward (y-axis) vs number of training samples (x-axis) for ORIGINAL and CONFOUNDED, compared to expert reward (mean and stdev over 5 runs). Also see Appendix D.

policies corresponding to various causal graphs (Sec 4.1). Then, we perform targeted interventions to efficiently search over the hypothesis set for the correct causal model (Sec 4.2).

4.1 Causal Graph-Parameterized Policy Learning

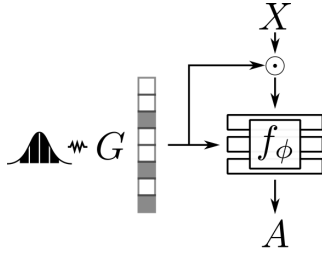


Figure 5: Graph-parameterized policy.

In this step, we learn a policy corresponding to each candidate causal graph. Recall from Sec 3 that the expert’s actions A are based on an unknown subset of the state variables $\{X_i\}_{i=1}^n$. Each X_i may either be a cause or not, so there are 2^n possible graphs. We parameterize the structure G of the causal graph as a vector of n binary variables, each indicating the presence of an arrow from X_k to A in Fig 2. We then train a single graph-parameterized policy $\pi_G(X) = f_\phi([X \odot G, G])$, where \odot is element-wise multiplication, and $[\cdot, \cdot]$ denotes concatenation. ϕ are neural network parameters, trained through gradient descent to minimize:

$$\mathbb{E}_G[\ell(f_\phi([X_i \odot G, G]), A_i)], \quad (1)$$

where G is drawn uniformly at random over all 2^n graphs and ℓ is a mean squared error loss for the continuous action environments and a cross-entropy loss for the discrete action environments. Fig 5 shows a schematic of the training time architecture. The policy network f_ϕ mapping observations X to actions A represents a mixture of policies, one corresponding to each value of the binary causal graph structure variable G , which is sampled as a bernoulli random vector.

In Appendix C, we propose an approach to perform variational Bayesian causal discovery over graphs G , using a latent variable model to infer a distribution over functional causal models (graphs and associated parameters)—the modes of this distribution are the FCMs most consistent with the demonstration data. This resembles the scheme above, except that instead of uniform sampling, graphs are sampled preferentially from FCMs that fit the training demonstrations well. We compare both approaches in Sec 5, finding that simple uniform sampling nearly always suffices in preparation for the next step: targeted intervention.

4.2 Targeted Intervention

Having learned the graph-parameterized policy as in Sec 4.1, we propose targeted intervention to compute the likelihood $\mathcal{L}(G)$ of each causal graph structure hypothesis G . In a sense, imitation learning provides an ideal setting for studying interventional causal learning: causal confusion presents a clear challenge, while the fact that the problem is situated in a sequential decision process where the agent can interact with the world provides a natural mechanism for carrying out limited interventions.

We propose two intervention modes, both of which can be carried out by interaction with the environment via the actions:

Expert query mode. This is the standard intervention approach applied to imitation learning: intervene on X^t to assign it a value, and observe the expert response A . This requires an interactive expert, as in DAgger [40], but requires substantially fewer expert queries than DAgger, both because: (i) the queries serve only to disambiguate among a relatively small set of valid FCMs, and (ii) we use disagreement among the mixture of policies in f_ϕ to query the expert efficiently in an active learning approach. We summarize this approach in Algorithm 1.

Policy execution mode. It is not always possible to query an expert. For example, for a learner learning to drive a car by watching a human driver, it may not be possible to put the human driver into dangerous scenarios that the learner might encounter at intermediate stages of training. In cases like these where we would like to learn from pre-recorded demonstrations alone, we propose to intervene indirectly by using environmental returns (sum of rewards over time in an episode) $R = \sum_t r_t$. The policies $\pi_G(\cdot) = f_\phi([\cdot \odot G, G])$ corresponding to different hypotheses G are executed in the environment and the returns R_G collected. The likelihood of each graph is proportional to the exponentiated returns $\exp R_G$. The intuition is simple: environmental returns contain information about optimal expert policies even when experts are not queryable. Note that we do not even assume access to per-timestep rewards as in standard reinforcement learning; just the *sum* of rewards for each completed run. As such, this intervention mode is much more flexible. See Algorithm 2.

Note that both of the above intervention approaches evaluate individual hypotheses in isolation, but the number of hypotheses grows exponentially in the number of state variables. To handle larger states, we infer a graph distribution $p(G)$, by assuming an energy based model with a linear energy $E(G) = \langle w, G \rangle + b$, so the graph distribution is $p(G) = \prod_i p(G_i) = \prod_i \text{Bernoulli}(G_i | \sigma(w_i/\tau))$, where σ is the sigmoid, which factorizes in independent factors. The independence assumption is sensible as our approach collapses $p(G)$ to its mode before returning it and the collapsed distribution is always independent. $E(G)$ is inferred from linear regression on the likelihoods. This process is depicted in Algorithms 1 and 2. The above method can be formalized within the reinforcement learning framework [25]. As we show in Appendix G, the energy-based model can be seen as an instance of soft Q-learning [14].

4.3 Disentangling Observations

In the above, we have assumed access to disentangled observations X^t . When this is not the case, such as with image observations, X^t must be set to a disentangled representation of the observation at time t . We construct such a representation by training a β -VAE [20, 16] to reconstruct the original observations. To capture states beyond those encountered by the expert, we train with a mix of expert and random trajectory states. Once trained, X^t is set to be the mean of the latent distribution produced at the output of the encoder. The VAE training objective encourages disentangled dimensions in the latent space [5, 6]. We employ CoordConv [26] in both the encoder and the decoder architectures.

5 Experiments

We now evaluate the solution described in Sec 4 on the five tasks (MountainCar, Hopper, and 3 Atari games) described in Sec 3.2. In particular, recall that CONFOUNDED performed significantly worse than ORIGINAL across all tasks. In our experiments, we seek to answer the following questions: **(1)** Does our targeted intervention-based solution to causal confusion bridge the gap between CONFOUNDED and ORIGINAL? **(2)** How quickly does performance improve with intervention? **(3)** Do both intervention modes (expert query, policy execution) described in Sec 4.2 resolve causal confusion? **(4)** Does our approach in fact recover the true causal graph?

In each of the two intervention modes, we compare two variants of our method: UNIF-INTERVENTION and DISC-INTERVENTION. They only differ in the training of the graph-parameterized mixture-of-

Algorithm 1 Expert query intervention

Input: policy network f_ϕ s.t. $\pi_G(X) = f_\phi([X \odot G, G])$
Initialize $w = 0, \mathcal{D} = \emptyset$.
Collect states \mathcal{S} by executing π_{mix} , the mixture of policies π_G for uniform samples G .
For each X in \mathcal{S} , compute disagreement score:
 $D(X) = \mathbb{E}_G[D_{KL}(\pi_G(X), \pi_{mix}(X))]$
Select $\mathcal{S}' \subset \mathcal{S}$ with maximal $D(X)$.
Collect state-action pairs \mathcal{T} by querying expert on \mathcal{S}' .
for $i = 1 \dots N$ **do**
 Sample $G \sim p(G) \propto \exp \langle w, G \rangle$.
 $\mathcal{L} \leftarrow \mathbb{E}_{s,a \sim \mathcal{T}}[\ell(\pi_G(s), a)]$
 $\mathcal{D} \leftarrow \mathcal{D} \cup \{(G, \mathcal{L})\}$
 Fit w on \mathcal{D} with linear regression.
end for
Return: $\arg \max_G p(G)$

Algorithm 2 Policy execution intervention

Input: policy network f_ϕ s.t. $\pi_G(X) = f_\phi([X \odot G, G])$
Initialize $w = 0, \mathcal{D} = \emptyset$.
for $i = 1 \dots N$ **do**
 Sample $G \sim p(G) \propto \exp \langle w, G \rangle$.
 Collect episode return R_G by executing π_G .
 $\mathcal{D} \leftarrow \mathcal{D} \cup \{(G, R_G)\}$
 Fit w on \mathcal{D} with linear regression.
end for
Return: $\arg \max_G p(G)$

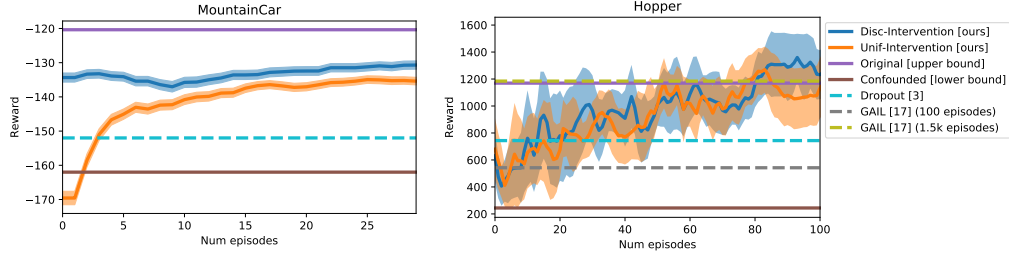


Figure 6: Reward vs. number of intervention episodes (policy execution interventions). Our methods UNIF-INTERVENTION and DISC-INTERVENTION bridge most of the causal confusion gap (between ORIGINAL (lower bound) and CONFOUNDED (upper bound), approaching ORIGINAL performance after tens of episodes. GAIL [17] (on Hopper) achieves this too, but after 1.5k episodes.

policies f_ϕ —while UNIF-INTERVENTION samples causal graphs uniformly, DISC-INTERVENTION uses the variational causal discovery approach mentioned in Sec 4.1, and described in detail in Appendix C.

Baselines. We compare our method against three baselines applied to the confounded state. DROPOUT trains the policy using Eq 3 and evaluates with the graph G containing all ones, which amounts to dropout regularization [51] during training, as proposed by Bansal et al. [3]. DAGGER [40] addresses distributional shift by querying the expert on states encountered by the imitator, requiring an interactive expert. We compare DAGGER to our expert query intervention approach. Lastly, we compare to Generative Adversarial Imitation Learning (GAIL) [17]. GAIL is an alternative to standard behavioral cloning that works by matching demonstration trajectories to those generated by the imitator during roll-outs in the environment. Note that the PC algorithm [24], commonly used in causal discovery from passive observational data, relies on the faithfulness assumption, which causes it to be infeasible in our setting. See Appendices B & C for details.

Environment	Pong	Enduro	UpNDown
ORIGINAL (upper bd)	15.0	39.5	80.9
CONFOUNDED (lower bd)	-4.0	30.5	24.8
ORIGINAL w/ VAE	12.3	36.7	54.5
COUNFOUNDED w/ VAE	-4.0	28.2	24.0
UNIF-INTERVENTION (ours)	11.6	32.4	66.3
DROPOUT [3]	-8.3	28.2	40.4

Table 2: Intervention by policy execution: Reward of the best models produced by our approach on Atari games. UNIF-INTERVENTION succeeds in getting rewards close to ORIGINAL w/ VAE, while the DROPOUT baseline only outperforms CONFOUNDED w/ VAE in UpNDown.

Intervention by policy execution. Fig 6 plots episode rewards versus number of policy execution intervention episodes for MountainCar and Hopper. The reward always corresponds to the current mode $\arg \max_G p(G)$ of the posterior distribution over graphs, updated after each episode, as described in Algorithm 2. In these cases, both UNIF-INTERVENTION and DISC-INTERVENTION eventually converge to models yielding similar rewards, which we verified to be the correct causal model i.e., true causes are selected and nuisance correlates left out. In early episodes on MountainCar, DISC-INTERVENTION benefits from the prior over graphs inferred in the variational causal discovery phase. However, in Hopper, the simpler UNIF-INTERVENTION performs just as well. DROPOUT does indeed help in both settings, as reported in Bansal et al. [3], but is significantly poorer than our approach variants. GAIL requires about 1.5k episodes on Hopper to match the performance of our approaches, which only need tens of episodes. Appendix F further analyzes the performance of GAIL. Standard implementations of GAIL do not handle discrete action spaces, so we do not evaluate it on MountainCar.

Experiments on Atari games are more computationally expensive, so we report results after a heuristically pre-selected number of episodes (1000). As described in Sec 4.3, we use a VAE to disentangle image states in Atari games to produce 30-D representations. Requiring the policy to utilize the VAE representation without end-to-end training does result in some drop in performance, as seen in Tab 1. However, causal confusion still causes a very large drop of performance even relative to the baseline VAE performance. As Tab 2 shows, UNIF-INTERVENTION indeed improves significantly over CONFOUNDED w/ VAE in all three cases, matching ORIGINAL w/ VAE on Pong and UpNDown, while the DROPOUT baseline only improves UpNDown. In our experiments thus far, GAIL fails to converge to above-chance performance on any of the Atari environments. These results show that our method successfully alleviates causal confusion within relatively few trials.

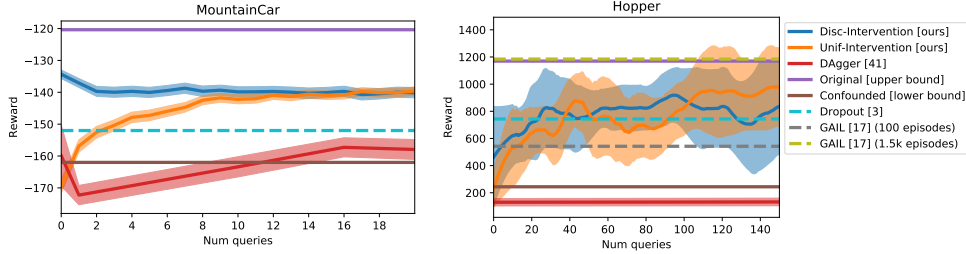


Figure 7: Reward vs. expert queries (expert query interventions). Our methods partially bridge the gap from CONFOUNDED (lower bd) to ORIGINAL (upper bd), also outperforming DAGGER [41] and DROPOUT [3]. GAIL [17] outperforms our methods on Hopper, but requires a large number of policy roll-outs (also see Fig 6 comparing GAIL to our policy execution-based approach).

Intervention by expert queries. Next, we perform direct intervention by querying the expert on samples from trajectories produced by the different causal graphs. In this setting, we can also directly compare to DAGGER [41]. Fig 7 shows results on MountainCar and Hopper. Both our approaches successfully improve over CONFOUNDED within a small number of queries. Consistent with policy execution intervention results reported above, we verify that our approach again identifies the true causal model correctly in both tasks, and also performs better than DROPOUT in both settings. It also exceeds the rewards achieved by DAGGER, while using far fewer expert queries. In Appendix E, we show that DAGGER requires hundreds of queries to achieve similar rewards for MountainCar and tens of thousands for Hopper. Finally, GAIL with 1.5k episodes outperforms our expert query interventions approach. Recall however from Fig 7 that this is an order of magnitude more than the number of episodes required by our policy intervention approach.

Once again, DISC-INTERVENTION only helps in early interventions on MountainCar, and not at all on Hopper. Thus, our method’s performance is primarily attributable to the targeted intervention stage, and the exact choice of approach used to learn the mixture of policies is relatively insignificant.

Overall, of the two intervention approaches, policy execution converges to better final rewards. Indeed, for the Atari environments, we observed that expert query interventions proved ineffective. We believe this is because expert agreement is an imperfect proxy for true environmental rewards.

Interpreting the learned causal graph.

Our method labels each dimension of the VAE encoding of the frame as a cause or nuisance variable. In Fig 8, we analyze these inferences in the Pong environment as follows: in the top row, a frame is encoded into the VAE latent, then for all nuisance dimensions (as inferred by our approach UNIF-INTERVENTION), that dimension is replaced with a sample from the prior, and new samples are generated.



Figure 8: Samples from (top row) learned causal graph and (bottom row) random causal graph. (See text)

In the bottom row, the same procedure is applied with a random graph that has as many nuisance variables as the inferred graph. We observe that in the top row, the causal variables (the ball and paddles) are shared between the samples, while the nuisance variables (the digit) differ, being replaced either with random digits or unreadable digits. In the bottom row, the causal variables differ strongly, indicating that important aspects of the state are judged as nuisance variables. This validates that, consistent with MountainCar and Hopper, our approach does indeed identify true causes in Pong.

6 Conclusions

We have identified a naturally occurring and fundamental problem in imitation learning, “causal confusion”, and proposed a causally motivated approach for resolving it. While we observe evidence for causal confusion arising in natural imitation learning settings, we have thus far validated our solution in somewhat simpler synthetic settings intended to mimic them. Extending our solution to work for such realistic scenarios is an exciting direction for future work. Finally, apart from imitation, general machine learning systems deployed in the real world also encounter “feedback” [45, 2], which opens the door to causal confusion. We hope to address these more general settings in the future.

Acknowledgments: We would like to thank Karthikeyan Shanmugam and Shane Gu for pointers to prior work early in the project, and Yang Gao, Abhishek Gupta, Marvin Zhang, Alyosha Efros, and Roberto Calandra for helpful discussions in various stages of the project. We are also grateful to Drew Bagnell and Katerina Fragkiadaki for helpful feedback on an earlier draft of this paper. This project was supported in part by Berkeley DeepDrive, NVIDIA, and Google.

References

- [1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.
- [2] Drew Bagnell. Talk: Feedback in machine learning, 2016. URL <https://youtu.be/XRSvz4U0po4>.
- [3] Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. *Robotics: Science & Systems (RSS)*, 2019.
- [4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Praseen Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [5] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β -vae. *arXiv preprint arXiv:1804.03599*, 2018.
- [6] Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.
- [7] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
- [8] Hal Daumé, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009.
- [9] Frederick Eberhardt. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3(2):81–91, 2017.
- [10] Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.
- [11] Alessandro Giusti, Jerome Guzzi, Dan Ciresan, Fang-Lin He, Juan Pablo Rodriguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jurgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca Gambardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 2016.
- [12] Olivier Goudet, Diviyani Kalainathan, Philippe Caillou, David Lopez-Paz, Isabelle Guyon, Michele Sebag, Aris Tritas, and Paola Tubaro. Learning functional causal models with generative neural networks. *arXiv preprint arXiv:1709.05321*, 2017.
- [13] Isabelle Guyon, Constantin Aliferis, Greg Cooper, André Elisseeff, Jean-Philippe Pellet, Peter Spirtes, and Alexander Statnikov. Design and analysis of the causation and prediction challenge. In *Causation and Prediction Challenge*, pages 1–33, 2008.
- [14] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.
- [15] David Heckerman, Christopher Meek, and Gregory Cooper. *A Bayesian Approach to Causal Discovery*, pages 1–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-33486-6. doi: 10.1007/3-540-33486-6_1. URL https://doi.org/10.1007/3-540-33486-6_1.
- [16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2017.
- [17] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

- [18] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):21, 2017.
- [19] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
- [20] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [21] Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2575–2583. Curran Associates, Inc., 2015.
- [22] Philipp Krähenbühl. Free supervision from video games. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2955–2964, 2018.
- [23] Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. *arXiv preprint arXiv:1703.09327*, 2017.
- [24] Thuc Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM transactions on computational biology and bioinformatics*, 2016.
- [25] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018.
- [26] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. *CoRR*, abs/1807.03247, 2018. URL <http://arxiv.org/abs/1807.03247>.
- [27] D. Lopez-Paz, R. Nishihara, S. Chintala, B. Schölkopf, and L. Bottou. Discovering causal signals in images. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, pages 58–66, Piscataway, NJ, USA, July 2017. IEEE.
- [28] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.
- [29] Marloes H Maathuis, Diego Colombo, Markus Kalisch, and Peter Bühlmann. Predicting causal effects in large-scale systems from observational data. *Nature Methods*, 7(4):247, 2010.
- [30] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
- [31] Jeffrey Mahler and Ken Goldberg. Learning deep policies for robot bin picking by simulating robust grasping sequences. In *Conference on Robot Learning*, pages 515–524, 2017.
- [32] Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. Causal inference via kernel deviance measures. *arXiv preprint arXiv:1804.04622*, 2018.
- [33] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- [34] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann L Cun. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pages 739–746, 2006.
- [35] Katharina Mülling, Jens Kober, Oliver Kroemer, and Jan Peters. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.
- [36] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [37] Jonas Peters, Joris M Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1): 2009–2053, 2014.
- [38] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.

- [39] Dean A Pomerleau. Alvin: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.
- [40] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 661–668, 2010.
- [41] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, 2011.
- [42] Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in cognitive sciences*, 3(6):233–242, 1999.
- [43] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.
- [44] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [45] David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in neural information processing systems*, pages 2503–2511, 2015.
- [46] Rajat Sen, Karthikeyan Shanmugam, Alexandros G Dimakis, and Sanjay Shakkottai. Identifying best interventions through online importance sampling. *arXiv preprint arXiv:1701.02789*, 2017.
- [47] Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. In *Advances in Neural Information Processing Systems*, pages 3195–3203, 2015.
- [48] Peter Spirtes. Introduction to causal inference. *Journal of Machine Learning Research*, 11 (May):1643–1662, 2010.
- [49] Peter Spirtes and Kun Zhang. Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, 2016.
- [50] Peter Spirtes, Clark N Glymour, Richard Scheines, David Heckerman, Christopher Meek, Gregory Cooper, and Thomas Richardson. *Causation, prediction, and search*. MIT press, 2000.
- [51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [52] Mark Steyvers, Joshua B Tenenbaum, Eric-Jan Wagenmakers, and Ben Blum. Inferring causal networks from observations and interventions. *Cognitive science*, 27(3):453–489, 2003.
- [53] Simon Tong and Daphne Koller. Active learning for structure in bayesian networks. In *IJCAI*, 2001.
- [54] Dequan Wang, Coline Devin, Qi-Zhi Cai, Philipp Krähenbühl, and Trevor Darrell. Monocular plan view networks for autonomous driving. *arXiv preprint arXiv:1905.06937*, 2019.
- [55] Yixin Wang and David M Blei. The blessings of multiple causes. *arXiv preprint arXiv:1805.06826*, 2018.
- [56] Bernard Widrow and Fred W Smith. Pattern-recognizing control systems, 1964.

A Expert Demonstrations

To collect demonstrations, we first train an expert with reinforcement learning. We use DQN [33] for MountainCar, TRPO [43] for Hopper, and PPO [44] for the Atari environments (Pong, UpNDown, Enduro). This expert policy is executed in the environment to collect demonstrations.

B Necessity of Correct Causal Model

Faithfulness: A causal model is said to be faithful when all conditional independence relationships in the distribution are represented in the graph.

We pick up the notation used in Sec 3.1, but for notational simplicity, we drop the time superscript for X , A , and Z when we are not reasoning about multiple time-steps.

Proposition 1. *Let the expert’s functional causal model be $(G^*, \theta_{G^*}^*)$, with causal graph $G^* \in \mathcal{G}$ as in Figure 2 and function parameters $\theta_{G^*}^*$. We assume some faithful learner $(\hat{G}, \theta_{\hat{G}})$, $\hat{G} \in \mathcal{G}$ that agrees on the interventional query:*

$$\forall X, A : p_{G^*, \theta_{G^*}^*}(A|do(X)) = p_{\hat{G}, \theta_{\hat{G}}}(A|do(X))$$

Then it must be that $G^ = \hat{G}$.²*

Proof. For graph G , define the index set of state variables that are independent of the action in the mutilated graph $G_{\bar{X}}$:

$$I_G = \{i | X_i \perp\!\!\!\perp_{G_{\bar{X}}} A\}$$

From the assumption of matching interventional queries and the assumption of faithfulness, it follows that: $I_{G^*} = I_{\hat{G}}$. From the graph, we observe that $I_G = \{i | (X_i \rightarrow A) \notin G\}$ and thus $G^* = \hat{G}$. \square

C Variational Causal Discovery

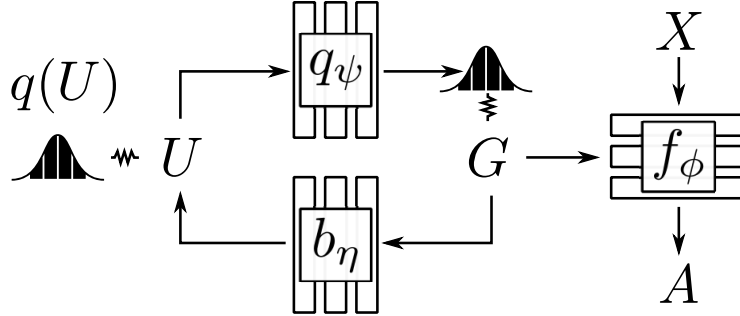


Figure 9: Training architecture for variational inference-based causal discovery as described in Appendix C. The policy network f_ϕ represents a mixture of policies, one corresponding to each value of the binary causal graph structure variable G . This variable in turn is sampled from the distribution $q_\psi(G|u)$ produced by an inference network from an input latent U . Further, a network b_η regresses back to the latent U to enforce that G should not be independent of U .

The problem of discovering causal graphs from passively observed data is called causal discovery. The PC algorithm [50] is arguably the most widely used and easily implementable causal discovery algorithm. In the case of Fig 2, the PC algorithm would imply the absence of the arrow $X_i^t \rightarrow A^t$, if the conditional independence relation $A^t \perp\!\!\!\perp X_i^t | Z^t$ holds, which can be tested by measuring the mutual information. However, the PC algorithm relies on *faithfulness* of the causal graph. That is, conditional independence must imply d-separation in the graph. However, faithfulness is easily violated in a Markov decision process. If for some i , X_i^t is a cause of the expert’s action A^t (the

²We drop time t from the superscript when discussing states and actions from the same time.

arrow $X_i^t \rightarrow A^t$ should exist), but X_i^t is the result of a deterministic function of Z^t , then always $A^t \perp\!\!\!\perp X_i^t | Z^t$ and the PC algorithm would wrongly conclude that the arrow $X_i^t \rightarrow A^t$ is absent.³

We take a Bayesian approach to causal discovery [15] from demonstrations. Recall from Sec 3 that the expert’s actions A are based on an unknown subset of the state variables $\{X_i\}_{i=1}^n$. Each X_i may either be a cause or not, so there are 2^n possible graphs. We now define a variational inference approach to infer a distribution over functional causal models (graphs and associated parameters) such that its modes are consistent with the demonstration data D .

While Bayesian inference is intractable, variational inference can be used to find a distribution that is close to the true posterior distribution over models. We parameterize the structure G of the causal graph as a vector of n correlated Bernoulli random variables G_k , each indicating the presence of a causal arrow from X_k to A . We assume a variational family with a point estimate θ_G of the parameters corresponding to graph G and use a latent variable model to describe the correlated Bernoulli variables, with a standard normal distribution $q(U)$ over latent random variable U :

$$\begin{aligned} q_\psi(G, \theta) &= q_\psi(G)[\theta = \theta_G] \\ &= \int q(U) \prod_{k=1}^n q_\psi(G_k | U)[\theta = \theta_G] dU \end{aligned}$$

We now optimise the evidence lower bound (ELBO):

$$\begin{aligned} \arg \min_q D_{KL}(q_\psi(G, \theta) | p(G, \theta | D)) &= \\ \arg \max_{\psi, \theta} \sum_i \mathbb{E}_{U \sim q(U), G_k \sim q_\psi(G_k | U)} & \quad (2) \\ [\log \pi(A_i | X_i, G, \theta_G) + \log p(G)] + \mathcal{H}_q(G) & \quad (3) \end{aligned}$$

Likelihood $\pi(A_i | X_i, G, \theta_G)$ is the likelihood of the observations X under the FCM (G, θ_G) . It is modelled by a single neural network $f_\phi([X \odot G, G])$, where \odot is the element-wise multiplication, $[\cdot, \cdot]$ denotes concatenation and ϕ are neural network parameters.

Entropy The entropy term of the KL divergence, \mathcal{H}_q , acts as a regularizer to prevent the graph distribution from collapsing to the maximum a-posteriori estimate. It is intractable to directly maximize entropy, but a tractable variational lower bound can be formulated. Using the product rule of entropies, we may write:

$$\begin{aligned} \mathcal{H}_q(G) &= \mathcal{H}_q(G|U) - \mathcal{H}_q(U|G) + \mathcal{H}_q(U) \\ &= \mathcal{H}_q(G|U) + I_q(U; G) \end{aligned}$$

In this expression, $\mathcal{H}_q(G|U)$ promotes diversity of graphs, while $I_q(U; G)$ encourages correlation among $\{G_k\}$. $I_q(U; G)$ can be bounded below using the same variational bound used in InfoGAN [7], with a variational distribution b_η : $I_q(U; G) \geq \mathbb{E}_{U, G \sim q_\psi} \log b_\eta(U|G)$. Thus, during optimization, in lieu of entropy, we maximize the following lower bound:

$$\mathcal{H}_q(G) \geq \mathbb{E}_{U, G \sim q} \left[- \sum_k \log q_\psi(G_k | U) + \log b_\eta(U | G) \right]$$

Prior The prior $p(G)$ over graph structures is set to prefer graphs with fewer causes for action A —it is thus a sparsity prior:

$$p(G) \propto \exp \sum_k [G_k = 1]$$

³More generally, faithfulness places strong constraints on the expert graph. For example, a visual state may contain unchanging elements such as the car frame in Fig 1, which are by definition deterministic functions of the past. As another example, goal-conditioned tasks must include a constant goal in the state variable at each time, which once again has deterministic transitions, violating faithfulness.

Optimization Note that G is a discrete variable, so we cannot use the reparameterization trick [20]. Instead, we use the Gumbel Softmax trick [19, 30] to compute gradients for training $q_\psi(G_k|U)$. Note that this does not affect f_ϕ , which can be trained with standard backpropagation.

The loss of Eq 3 is easily interpretable independent of the formalism of variational Bayesian causal discovery. A mixture of predictors f_ϕ is jointly trained, each paying attention to diverse sparse subsets (identified by G) of the inputs. This is related to variational dropout [21]. Once this model is trained, $q_\psi(G)$ represents the hypothesis distribution over graphs, and $\pi_G(x) = f_\phi([x \odot G, G])$ represents the imitation policy corresponding to a graph G . Fig 9 shows the architecture.

Usage for Targeted Interventions In our experiments, we also evaluate the usefulness of causal discovery process to set a prior for the targeted interventions described in Sec 4.2. In Algorithm 1 and 2, we implement this by initializing $p(G)$ to the discovered distribution (rather than uniform).

D Additional Results: Diagnosing Causal Confusion

In Fig 10 we show the causal confusion in several environments. We observe that while training and validation losses for behavior cloning are frequently near-zero for both the original and confounded policy, the confounded policy consistently yields significantly lower reward when deployed in the environment. This confirms the causal confusion problem.

E DAgger with many more interventions

In the main paper, we showed that DAgger performed poorly with equal number of expert interventions as our method. How many more samples does it need to do well?

The results in Fig 11 show that DAgger requires hundreds of samples before reaching rewards comparable to the rewards achieved by a non-DAgger imitator trained on the original state.

F GAIL Training Curves

In Figure 12 we show the average training curves of GAIL on the original and confounded state. Error bars are 2 standard errors of the mean. The confounded and original training curve do not differ significantly, indicating that causal confusion is not an issue with GAIL. However, training requires many interactions with the environment.

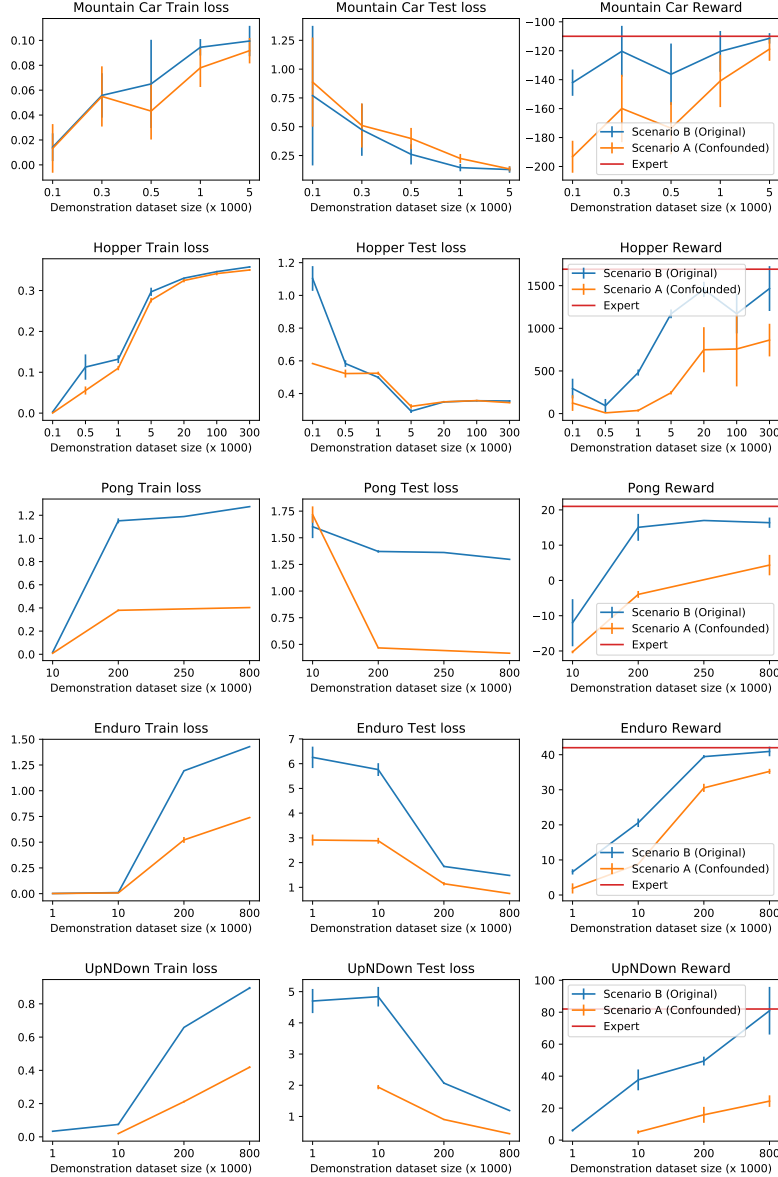


Figure 10: An expanded version of Fig 4 in the main paper, demonstrating diagnosis of the causal confusion problem in three settings. Here, the final reward, shown in Fig 4 is shown in the third column. Additionally, we also show the behavior cloning training loss (first column) and validation loss (second column) on trajectories generated by the expert. The x-axis for all plots is the number of training examples used to train the behavior cloning policy.

G Intervention Posterior Inference as Reinforcement Learning

Given a method of evaluating the likelihood $p(\mathcal{O}|G)$ of a certain graph G to be optimal and a prior $p_0(G)$, we wish to infer the posterior $p(G|\mathcal{O})$. The number of graphs is finite, so we can compute this posterior exactly. However, there may be very many graphs, so that impractically many likelihood evaluations are necessary. Only noisy samples from the likelihood can be obtained, as in the case of intervention through policy execution, where the reward is noisy, this problem is exacerbated.

If on the other hand, a certain structure on the policy is assumed, the sample efficiently can be drastically improved, even though policy can no longer be exactly inferred. This can be done in

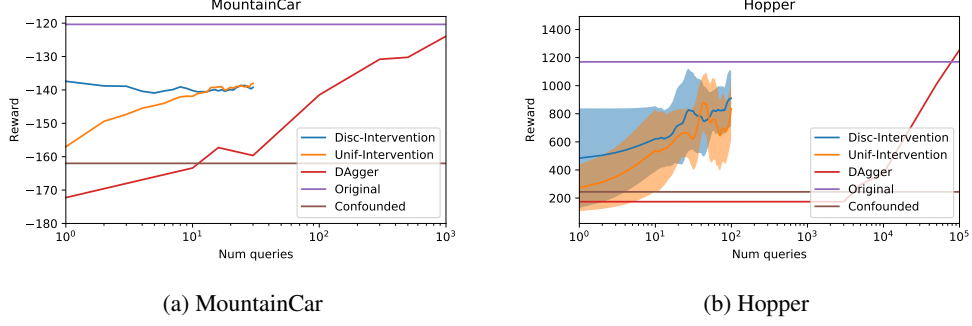


Figure 11: DAGger results trained on the confounded state.

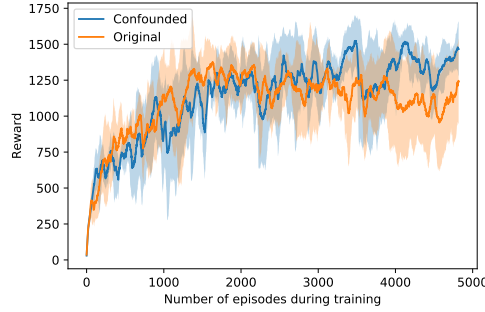


Figure 12: Rewards during GAIL training.

the framework of Variational Inference. For a certain variational family, we wish to find, for some temperature τ :

$$\pi(G) = \arg \min_{\pi(G)} D_{KL}(\pi(G) || p(\mathcal{O}|G)) \quad (4)$$

$$= \arg \min_{\pi(G)} \mathbb{E}_{\pi} [\log p(\mathcal{O}|G) + \log p_0(G)] + \tau \mathcal{H}_{\pi}(G) \quad (5)$$

The variational family we assume is the family of independent distributions:

$$\pi(G) = \prod_i \pi_i(G_i) = \prod_i \text{Bernoulli}(G_i | \sigma(w_i/\tau)) \quad (6)$$

Eq 5 can be interpreted as a 1 step entropy-regularized MDP with reward $\tilde{r} = \log p(\mathcal{O}|G) + \log p_0(G)$ [25]. It can be optimized through a policy gradient, but this would require many likelihood evaluations. More efficient is to use a value based method. The independence assumption translates in a linear Q function: $Q(G) = \langle w, G \rangle + b$, which can be simply learned by linear regression on off-policy pairs (G, \tilde{r}) . In Soft Q-Learning [14] it is shown that the policy that maximizes Eq 5 is $\pi(G) \propto \exp Q(G)/\tau$, which can be shown to coincide in our case with Eq 6:

$$\begin{aligned} \pi(G) &= \frac{\exp(\langle w, G \rangle + b)/\tau}{\sum_{G'} \exp(\langle w, G' \rangle + b)/\tau} \propto \prod_i \exp(w_i G_i / \tau) \\ \Rightarrow \pi(G) &= \prod_i \frac{\exp(w_i G_i / \tau)}{1 + \exp w_i / \tau} = \prod_i \text{Bernoulli}(G_i | \sigma(w_i/\tau)) \end{aligned}$$