Can Active Sampling Reduce Causal Confusion in Offline Reinforcement Learning?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Causal confusion is a phenomenon where an agent learns a policy that reflects *imperfect spurious correlations* in the data. The resulting causally confused behaviors may appear desirable during training but may fail at deployment. This problem gets exacerbated in domains such as robotics with potentially large gaps between open- and closed-loop performance of an agent. In such cases, a causally confused model may appear to perform well according to open-loop metrics but fail catastrophically when deployed in the real world. In this paper, we conduct the first study of causal confusion in offline reinforcement learning and hypothesise that selectively sampling data points that may help disambiguate the underlying causal mechanism of the environment may alleviate causal confusion. To investigate this hypothesis, we consider a set of simulated setups to study causal confusion and the ability of active sampling schemes to reduce its effects. We provide empirical evidence that random and active sampling schemes are able to consistently reduce causal confusion as training progresses and that active sampling is able to do so more efficiently than random sampling.

1 Introduction

Offline learning offers opportunities to scale reinforcement learning to domains where offline data is plentiful but online interaction with the environment is costly. The fundamental challenge of offline reinforcement learning is to identify cause and effect of actions from a fixed dataset, which is often intractable. In the absence of online interactions, our hope is that the dataset covers a uniform distribution of an exhaustive set of plausible scenarios. This is often not the case in datasets for robotic control, which are long-tailed and often contain only a handful of samples for rare (and informative) events. Causal confusion occurs when agents misinterpret the underlying causal mechanisms of the environment and erroneously associate certain actions or states with a given reward. For example, if an agent happens to simultaneously observes independent events X and Y in its environment whenever it receives a reward R, and R causally depends on Y but not on X, the agent may attribute the reward R to X and Y occurring jointly even though R may be independent of Y. Problematically, if the spurious correlation between Y and R observed at training time ceases to hold at deployment time, a causally-confused model may show a significant deterioration in performance. Often, spurious correlations are not *perfectly* held in offline data, but optimisation schemes like mini-batched gradient descent can still produce models that latch onto them since they help in optimising the training loss. In this paper, we explore whether causal confusion in offline reinforcement learning from datasets exhibiting causal ambiguity can be alleviated by random or active sampling. We provide empirical evidence that random and active sampling schemes are able to consistently reduce causal confusion and that active sampling is able to do so more efficiently than random sampling.

Submitted to the NeurIPS Causal ML for Real-World Impact Workshop (CML4Impact 2022). Do not distribute

2 Related Work

Causal Confusion in Supervised Learning. Several works in imitation learning have proposed solutions to mitigate causal confusion, which was first defined in [de Haan et al., 2019]. Wen et al. [2020] proposes adversarial training to prune out any *known* sources of spurious correlations from the policy's representation, for instance, the previous control commands given to a robot; Wen et al. [2021] propose loss-reweighting of datapoints based on the loss of a model trained with just the spurious correlates as the input; OREO [Park et al., 2021] regularises the model's representation to be invariant to any individual object being dropped out in a scene. Lee et al. [2022] propose training a diversified ensemble in the case when *perfect spurious correlations* exist in the data and later select from these hypotheses based on validation data. Causal Confusion has also recently been studied in reward-learning from preferences [Tien et al., 2022], where spurious correlations can be drawn between a human evaluator's preferences and certain actions or parts of the state space.

Ensemble Models in RL. Ensembles have been studied extensively to guide exploration in online RL [Osband et al., 2016] [Lee et al., 2021], and recently to construct adaptive pessimism constraints in offline RL, to disincentivise uncertain actions from having high estimated returns. Recent work [An et al., 2021] showed that increasing the size and diversity of the ensembled critic in Soft-Actor-Critic [Haarnoja et al., 2018] performs competitively with state-of-the-art offline RL algorithms. However prior work hasn't explored how the uncertainty from ensembles could be used to sample transitions in RL. Prioritised replay [Schaul et al., 2015] is a sampling scheme based on the TD-error of transitions, that was proposed in off-policy RL but hasn't been studied in offline RL.

AI Alignment. AI alignment seeks to align the behavior of agents with the intentions of their creators by investigating the incentives behind demonstrated tasks. Recent work on *Goal Misgeneralisation* [Langosco et al., 2022] explores how online RL agents in Procgen [Cobbe et al., 2019] can get confused about the goal they're pursuing since those goals co-occur with irrelevant artifacts in the environment most of the time. In this case the specification is correct, but the agent still pursues an unintended objective (as opposed to poor reward definitions that predictably lead to reward hacking). We build upon an environment introduced in this work to collect data for reproducing the phenomenon of causal confusion in offline RL.

3 Alleviating Causal Confusing in Offline RL via Active Sampling

3.1 Offline Reinforcement Learning

Offline RL algorithms aim to learn an optimal policy along with estimates of the value (or Q-value) function from a dataset of transitions $\mathcal{D} = \{(\mathbf{s}, \mathbf{a}, r, \mathbf{s}')\}$ collected by a behaviour policy π_β .

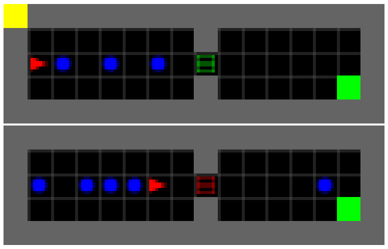
Conservative Q-Learning. For our experiemnts we choose CQL [Kumar et al., 2020] for it's simplicity and competitive performance. The CQL objective, which combines the standard TD-error of Q-learning with a penalty constraining deviations from the behaviour policy, is defined as:

$$\mathcal{L}_{\text{critic}}^{\text{CQL}}(\theta) = \frac{1}{2} \mathop{\mathbb{E}}_{(s,a,s')\sim\mathcal{D}} \left[(Q_\theta - \mathcal{B}^\pi Q_{\bar{\theta}})^2 \right] + \alpha_0 \mathop{\mathbb{E}}_{s\sim\mathcal{D}} \left[\log \sum_a \exp Q(s,a) - \mathop{\mathbb{E}}_{a\sim\pi_\beta} [Q(s,a)] \right], \quad (1)$$

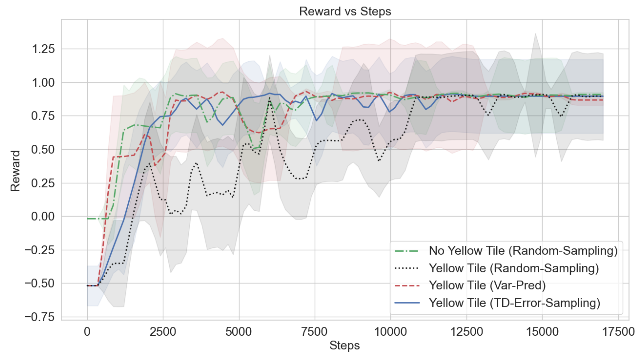
where the Bellman operator $\mathcal{B}^\pi Q = r + \gamma P^\pi Q$, and P^π is the transition matrix coupled with the policy π. We model the uncertainty of the learned Q-function parameterised by θ by ensembling the model and training on identical transitions across the ensemble members,with their own corresponding targets ($\bar{\theta}$) as proposed in Ghasemipour et al. [2022]

3.2 Active Sampling

The focus of this work is on experimenting with data-sampling strategies without making any modifications to the objective. Algorithm 1 describes the CQL setup with active-sampling of transitions, where the modifications from vanilla random-sampling are highlighted in blue. We study the following uncertainty-based and loss-based data acquisition schemes:



(a) **Top:** The leading vehicle is static and the top-left tile flashes yellow since the leading vehicle is static. **Bottom:** The agent is in front of a red light, and the top-left tile isn't yellow since the leading vehicle isn't static or blocked.



(b) Random sampling takes 4x gradient steps to recover the correct solution compared to active sampling (Variance and TD-Error-based) when both are trained on data with the spurious yellow tile.

Figure 1: Traffic-world environment.

Uncertainty about the greedy action (variance-based): The Q-values of different ensemble members could have arbitrary numerical offsets but still be equivalent, due to bootstrapping. Instead, we estimate the uncertainty of actions by computing the variance of their advantage over the ensemble, where the advantage of an action a^* for a Q-learner can be written as follows:

$$A^\pi(s,a^*) = Q^\pi(s,a^*) - V^\pi(s) \approx Q^\pi(s,a^*) - \sum_a \left[Q(s,a) \cdot \frac{e^{(Q(s,a)}}{\sum_{a'} e^{Q(s,a')}} \right] \tag{2}$$

TD-Error (loss-based): Based on the Temporal Difference error similar to Prioritised Experience Replay [Schaul et al., 2015]

In practice, computing the acquisition scores over all the transitions in the dataset can be very expensive and redundant since high-error or high-uncertainty points will likely stay informative for a short window of subsequent gradient steps. We thus recompute the scores after every n gradient steps, and vary n as a hyper-parameter in our experiments.

4 Experiments

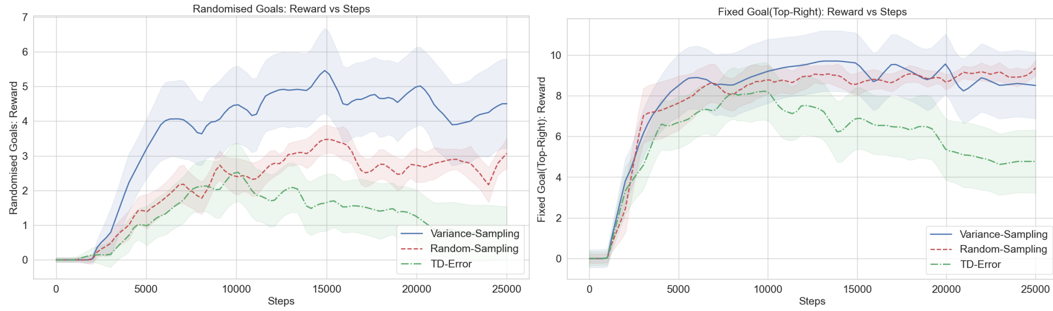
We investigate the following questions: (1) Can causal confusion be consistently observed in CQL when sampling transitions randomly from a long-tailed demonstration dataset? (2) Does active-sampling based on the (implicit) policy's uncertainty or loss help? (3) What is the computation time involved with each of these acquisition schemes?

Algorithm 1 Conservative Q-Learning (+ active-sampling)

```
Initialise ensemble Q-function Q_θ, n_ep=epochs, d_sz=dataset size, b_sz=batch size, T=steps-per-epoch.
for epoch e in {1, ..., n_ep} do
    for step t in {1, ..., T} do
        compute scores acq_i over D_train = [s_i, a_i]_{i=1}^{d_sz} according to the acquisition function
        acq_i = acq_i / Σ_{j=1}^{d_sz} acq_j (normalise scores)
        sample batch B = [s_i, a_i, s'_i, r_i]_{i=1}^{b_sz} from D_train ~ multinomial(acq)
        Train the Q-function on D_train using objective from Equation (1)
    end for
end for
```

4.1 Illustrative Example: Traffic-World

The autonomous driving literature cites many examples where models training on large datasets are very performant but exhibit causal confusion on the tail cases of their operational domain, for instance: (1) models stopping at pedestrian crossings regardless of whether a pedestrian is present or not since the two often co-occur; (2) self-driving agents that simply try to *cruise* if they know their current speed since expert driving datasets contain cruising behaviour in a large fraction of each trajectory. We build on the environment proposed in [Anonymous, 2021] to construct a gridworld (shown in Figure 1a), where an agent (red triangle) starts at the leftmost point in a row behind leading vehicles (blue circles), and needs to cross a traffic light to reach a goal location (green square) on the right side of the grid. We collect data such that the probability of the traffic light turning red becomes lower as the agent approaches it, and so the data distribution contains (1) mostly episodes where the light is green, (2) some episodes where the traffic



(a) Performance with randomly-sampled goals. (b) Performance with goals at the top-right.

Figure 2: Agents trained on a dataset containing 6000 episodes with a fixed goal and 200 episodes with a randomly sampled goal in Maze. We see that random-sampling and active-sampling perform similarly on the fixed goal evaluation environment (right), but the active-sampling variants achieve higher reward in the environments with randomly sampled goals. This verifies that the model is not just performing well in one of the two kinds of environments, is not constrained by capacity, and the reason behind the lower performance of random-sampling in this case is causal confusion.

light is red and the agent is waiting behind the vehicle in front (referenced here onward as the leading vehicle), and (3) only a couple of episodes where the light turns red with the agent at the front of the traffic queue. In this setup, the agent could just learn to follow the leading vehicle, instead of learning traffic light rules. To test causal confusion explicitly here, we introduce a related spurious correlate: a flashing yellow tile at the top left of the grid, that is yellow whenever the leading vehicle is stopped or blocked, and grey otherwise. The agent could follow this as an indicator of whether to stop or go ahead, and this policy would be correct for 98% of the data points. Figure 1b shows the training curves of CQL agents trained with randomly-sampled data, with and without the yellow tile present in images in the dataset - we see that the performance of the former agent degrades and it takes 4x the number of gradient steps to converge to the solution of the latter agent which is trained without the spurious correlate present. Also shown are the active sampling variants trained with the spurious yellow tile, which perform very similarly to random-sampling when the spurious correlate is not present.

4.2 Generalization in Offline RL: Procgen

The Maze environment in Procgen [Cobbe et al., 2019] defines a navigation task where the agent starts at the bottom left in the maze and receives a reward of +10 upon successfully reaching the goal which is sampled at any valid location in the maze. [Langosco et al., 2022] recently showed that an agent trained on a series of environments with the goal always at the top-right will be causally confused about the source of the reward. It will still navigate to the top-right even when the goal is sampled elsewhere. We generate a skewed *mixture* dataset containing mostly epsiodes where the goal is sampled at the top-right, and a few episodes where the goal is sampled randomly. Further details about the setup described in the Appendix. Figure 2b shows the evaluation performance of random and active-sampling agents trained on the *mixture* dataset, when goals are sampled randomly in the evaluation environment. We plot the computation time for the random and active-sampling variants in Figure 3 in the Appendix. Qualitative evaluations show that agents which achieve lower reward still successfully navigate to the top-right corner of the maze.

5 Conclusions

In this paper we designed preliminary setups to study causal confusion in offline RL, which occurs when a policy is learnt with random sampling of data from a skewed offline dataset. We designed uncertainty-based and loss-based data sampling baselines to selectively sample transitions for training, and saw promising evidence that active sampling can recover a less causally-confused model in significantly fewer training steps as compared to random-sampling. An interesting line of future work would be to scale this up to larger benchmarks, and extend this analysis to the case when acquisition scores for active sampling aren't computed for all the data at once, instead maintaining an approximation through running scores as is done in [Schaul et al., 2015].

References

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL `https://openreview.net/forum?id=ZUvaSolQZh3`.

Anonymous. Resolving causal confusion in reinforcement learning via robust exploration. In *Self-Supervision for Reinforcement Learning Workshop - ICLR 2021*, 2021. URL `https://openreview.net/forum?id=DKCXncD4Xtq`.

Karl Cobbe, Christopher Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. *arXiv preprint arXiv:1912.01588*, 2019.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/947018640bf36a2bb609d3557a285329-Paper.pdf`.

Seyed Kamyar Seyed Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline RL through ensembles, and why their independence matters., 2022. URL `https://openreview.net/forum?id=wQ7RCayXUSl`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/haarnoja18b.html`.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf`.

Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12004–12019. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/langosco22a.html`.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6131–6141. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/lee21g.html`.

Yoonho Lee, Huaxiu Yao, and Chelsea Finn. Diversify and disambiguate: Learning from underspecified data. 2022.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/8d8818c8e140c64c743113f563cf750f-Paper.pdf`.

Jongjin Park, Younggyo Seo, Chang Liu, Li Zhao, Tao Qin, Jinwoo Shin, and Tie-Yan Liu. Object-aware regularization for addressing causal confusion in imitation learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3029–3042. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/17a3120e4e5fbdc3cb5b5f946809b06a-Paper.pdf`.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized Experience Replay. *arXiv e-prints*, art. arXiv:1511.05952, November 2015.

Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca Dragan, and Daniel S. Brown. A study of causal confusion in preference-based reward learning. In *ICML 2022: Workshop on Spurious Correlations, Invariance and Stability*, 2022. URL `https://openreview.net/forum?id=WaZZ0Sw9fWf`.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *NeurIPS*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1b113258af3968aaf3969ca67e744ff8-Abstract.html`.

Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-focused visual imitation learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11123–11133. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/wen21d.html`.

A Appendix

A.1 Implementation

All our environments use a discrete action space. Therefore we build our method on top of the double-DQN implementation similar to the original CQL paper. As stated in section 1, we use ensembles of Q networks and at evaluation time, we average the Q-value outputs of the ensemble, and select the action with the maximum Q-value. In other place where we need to do inference (for instance: to compute Q-values for the conservative loss) we similarly take the mean across the ensemble.

A.2 Code and Data

We will release our code, data and pretrained models once the work is uploaded online. The code repository will also contain code to reproduce all the figures in this work.

A.3 Data Collection

1. Traffic-World: To collect data for Offline RL, we trained a PPO agent on a slightly modified version of the Traffic-world environment, with reward shaping on the environment, to incentivise the agent to reach the goal since this could is a hard exploration environment (there is the potential to receive many negative rewards before receiving a positive reward, and without reward shaping the PPO agent just learns to toggle in-place till the episode ends to avoid negative penalties).
2. Maze: We use the Impala-based PPO agent trained in [Langosco et al., 2022] for 200M steps to collect the expert trajectories on 6000 episodes of epsides with randomised goals and 200 episodes of episodes with fixed goals.

A.4 Hyper-parameters

CQL: We conduct a grid search over the learning rate and conservative penalty coefficient (α). We use gradient clipping with the norm varied between 3,5,7.

Active Sampling: We kept all the hyper-parameters the same as random sampling (batch size, learning rate, α). The parameters related to active sampling are

1. n: the number of gradient steps with stale scores we take before we recompute acquisition scores on the data.
2. the ensemble size which we set to 3, and keep constant across the active and random-sampling variants for a fair comparison.

A.5 Computational Cost

Figure 3 shows a scatterplot for the wallclock times to achieve highest reward across different active and random baselines. It also plots the time needed for active sampling variants to achiebve the best reward that random sampling achieves (denoted as Variance-par-Random and TD-par-Random in the

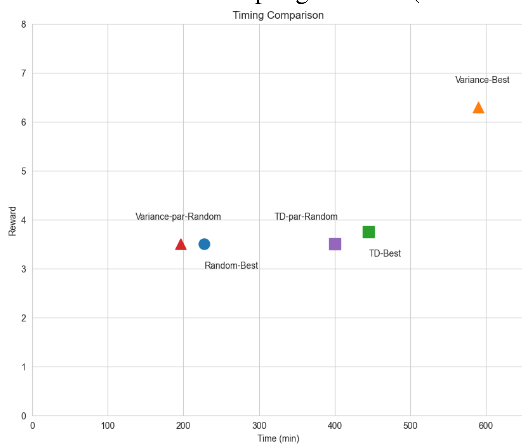


Figure 3: Timing Comparison for different sampling schemes on the Procgen-Maze benchmark plotted as reward achieved versus wallclock time in minutes.