A Study of Causal Confusion in Preference-Based Reward Learning

Anonymous Authors[1]

Abstract

While there is much empirical and theoretical analysis of causal confusion and reward gaming behaviors in reinforcement learning and behavioral cloning approaches, we provide the first systematic study of causal confusion in the context of learning reward functions from preferences. We identify a set of three benchmark domains where we observe causal confusion when learning reward functions from offline datasets of pairwise trajectory preferences: a simple reacher domain, an assistive feeding domain, and an itch-scratching domain. To gain insight into this observed causal confusion, we perform a sensitivity analysis on the effect of different factors—the reward model capacity and feature dimensionality—on the robustness of rewards learned from preferences. We find evidence that learning rewards from preferences is highly sensitive and non-robust to spurious features and increasing model capacity.

1. Introduction

Preference-based reward learning (Wirth et al., 2017; Sadigh et al., 2017; Christiano et al., 2017; Brown et al., 2020a) is a well-studied technique for learning from pairwise preferences or rankings. and holds the potential of allowing AI systems to learn specifications for tasks without requiring a human to write down an explicit reward function and to adapt the AI system's behavior to individual preferences and needs However, recent anecdotal evidence suggests that these methods are prone to learning rewards that overfit to spurious correlations in the data, especially when learning from limited numbers of preferences (Christiano et al., 2017; Ibarz et al., 2018; Javed et al., 2021). While the effects of reward misspecification have recently been studied in the context of reinforcement learning agents that optimize a proxy reward function (Pan et al., 2022) and behavioral cloning approaches that directly mimic an expert (De Haan et al., 2019; Zhang et al., 2020; Swamy et al., 2022), there has not been a systematic study of causal confusion when learning reward functions.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the SCIS workshop.

(a) Reacher (b) Feeding (c) ScratchItch

Figure 1. We propose and study three benchmark environments in which performing reward function learning from pairwise trajectory preferences leads to causal confusion over the true reward.

As an example of the type of causal confusion we study in this paper, consider the assistive feeding task in Figure 1b. Note that successful robot executions will cause the spoon to make contact with the patient's mouth, applying small amounts of force, whereas failed executions may not make any contact. We find that a preference-based learning approach will often miss the correct causal relationship between "contact" and "feeding", leading the robot to optimize a policy that seeks any kind of contact with the patient, including contact with the patient's torso or head.

We provide the following contributions: (1) We identify a set of 3 robot preference learning benchmarks that exhibit causal confusion when learning reward functions from preferences. (2) We perform the first systematic study of causal confusion in preference-based reward learning by varying the feature-space dimensionality and the capacity of the reward function model. (3) We identify spurious features and model capacity as the main sources of causal confusion when learning rewards from pairwise trajectory preferences.

2. Reward Learning from Preferences

We model the environment as a finite horizon MDP (Puterman, 2014), with state space $\mathcal{S}$, action space $\mathcal{A}$, horizon T, and reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. We assume that the reward function is unobserved and must be learned. The

reward function is learned from preferences over trajectories where, using the popular Bradley-Terry model (Bradley & Terry, 1952), the probability a trajectory, τ_B, is preferred over another trajectory, τ_A, is given by

$$P(\tau_A \prec \tau_B) = \frac{\exp(r(\tau_B))}{\exp(r(\tau_A)) + \exp(r(\tau_B)}, \qquad (1)$$

where $r(\tau) = \sum_{(s,a)\in\tau} r(s,a)$ and where we define a trajectory, τ, by a sequence of state-action pairs: $\tau = (s_0, a_0, \ldots, s_T, a_T)$.

To learn a reward function from preferences, we assume access to a set of pair-wise preference labels, $\mathcal{P}$, over trajectories, $\tau_1, \ldots, \tau_N$, where $(i,j) \in \mathcal{P}$ implies that $\tau_i \prec \tau_j$. We then optimize a reward function $r_\theta : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, parameterized by θ that maximizes the likelihood:

$$\mathcal{L}(\theta) = \prod_{(i,j)\in\mathcal{P}} \frac{\exp(r_\theta(\tau_j))}{\exp(r_\theta(\tau_i)) + \exp(r_\theta(\tau_j))}. \qquad (2)$$

3. Environments for Preference Learning

We identify a set of robot learning benchmarks that exhibit causal confusion when learning reward functions from preferences. In ***Reacher*** (Brockman et al., 2016) (Figure 1a) the goal is to move an end effector to a desired goal location. In ***Feeding*** (Erickson et al., 2020) (Figure 1b) the goal is to feed the human using a spoon carrying pieces of food. Finally, in ***Itch Scratching*** (Erickson et al., 2020) (Figure 1c) the goal is to repeatedly scratch a desired itch location on the human's arm.

Each domain has a predefined "true" reward function r. This enables us to create synthetic demonstrations and preference labels. Note that while we use the ground-truth reward function for obtaining preference labels ($r(\tau_1) < r(\tau_2) \implies \tau_1 \prec \tau_2$), we assume no access to this reward function during policy learning, but rather seek to learn a policy that obeys a users preferences by first learning a model r_θ of the true reward function from preference labels $\mathcal{P}$ and then running RL on the learned reward function. We can then evaluate the learned policy on the true reward function r.

To facilitate reproducibility and encourage future research on causal reward confusion, we will open-source our code and training datasets. This combination of domains and training data forms the first set of benchmarks for studying causal confusion when learning reward functions.

4. Evidence of Causal Confusion

To demonstrate that each of these domains exhibits causal confusion, we show that learning a reward function from preferences followed by policy optimization using the learned reward leads to behavior that performs poorly under the true reward function (unobserved, but used to provide synthetic preferences).

Table 1. **Empirical evidence of causal confusion.** We compare policies optimized with a reward learned from preferences (PREF) against policies optimized with the true reward (GT). We also include the special case of PREF where the input features contain enough information to perfectly explain the preferences (FULL). (For Reacher, this means learning a reward based on state-action pairs, since preferences are based on distance to target and action norm.) Despite high pairwise classification accuracy, the policy performance achieved by PREF under the true reward is very low compared with GT. However, *the reward learned from preferences consistently prefers the PREF policy over the GT policy*.

	Pref. Acc.		Policy Perf.	
Domain	Train	Val	Learned	True
Reach (full)	0.883	0.868	**13.415**	-17.720
Reach (GT)	——	——	6.410	**-6.701**
Reach (pref)	0.889	0.882	**13.593**	-17.972
Reach (GT)	——	——	-4.143	**-6.701**
Feed (pref)	0.811	0.812	**28.952**	-139.482
Feed (GT)	——	——	14.217	**97.295**
Itch (pref)	0.895	0.897	**39.070**	-41.940
Itch (GT)	——	——	15.258	**269.231**

Synthetic Preference Generation To enhance scalability and reproducibility, we generate a large number of trajectory preferences by using a pretrained RL policy (trained using the true reward provided with each environment) and then injecting noise into the policy. See Appendix C for details.

Causal Confusion Results In Table 1, we show both the pairwise classification accuracy on the train and validation sets as well as the subsequent performance of the resulting learned policies after training on 2,000 pairwise preference labels. Surprisingly, we find that despite having high validation accuracy on distinguishing between better or worse trajectories, the learned reward functions do not lead to correct behavior in the learned policy (as indicated by the low cumulative reward values), even when given large amounts of pairwise preferences and when the preferences can be perfectly inferred from the input features (ie., they are *fully-observable*). In addition to this, we observe that the learned reward function consistently assigns higher rewards to the policy trained from preference learning rather than the policy trained on the ground truth reward; these results provide strong empirical evidence of causal confusion. Figure 5 displays examples of the learned policies.

5. Factors that May Lead to Causal Confusion

Our results in the previous section demonstrate strong evidence of causal confusion during reward learning. In partic-

Table 2. **Privileged features.** Reacher's action norm feature is included whenever preferences are said to be *fully-observable*.

ENVIRONMENT	PRIVILEGED FEATURES
REACHER	DISTANCE TO TARGET ACTION NORM
FEEDING	DISTANCE TO MOUTH NUM. FOOD PARTICLES IN MOUTH NUM. FOOD PARTICLES ON FLOOR
SCRATCHITCH	DISTANCE TO ITCH TARGET FORCE APPLIED AT TARGET

ular, we found that preference-based reward learning fails even when using an expressive neural network to learn the reward and even when given large numbers of pairwise preferences—for comparison, prior work showed successful reward learning for Atari games and MuJoCo locomotion tasks using less than 300 pairwise preferences (Brown et al., 2019). In the remainder of this paper, we systematically vary different aspects of preference-based reward learning to gain insight into the following questions: How does (1) the choice of observation feature space and (2) the model capacity of the reward-learning network affect causal confusion? See Appendix F for an analysis of training data's effects on causal confusion.

5.1. The Choice of Observation Feature Space

We study reward learning when hand-crafted (causal) features are combined with raw observation features. Surprisingly, the concatenation of the two—despite having strictly more information—leads to worse reward learning results than using hand-crafted features alone. This type of causal misidentification mirrors previous results shown in the context of behavioral cloning (De Haan et al., 2019).

Hand-crafted Privileged Features: Based on the results in Section 4, preference-learning does not work well on the given observation spaces of our three benchmark domains. Thus, we use a set of hand-designed "privileged" features, as defined in Table 2, that directly correspond to the ground truth reward. We then train a linear reward model on these features. We find that removing the bias incentivizes more desirable behavior. For more details, see Appendix D.

Given only causal features for the ground truth reward, preference learning is generally able to successfully learn weights over these features, leading to a good RL policy. We found that Feeding achieves a mean cumulative reward of **132.18** and task success rate of **99.7%**, and Itch Scratching achieves a mean cumulative reward of **205.12** and task success rate of **90.7%**. Reacher achieves a cumulative reward of -18.53, which increases to **-12.13** (-3.75 being considered good) when preferences are made fully-observable.

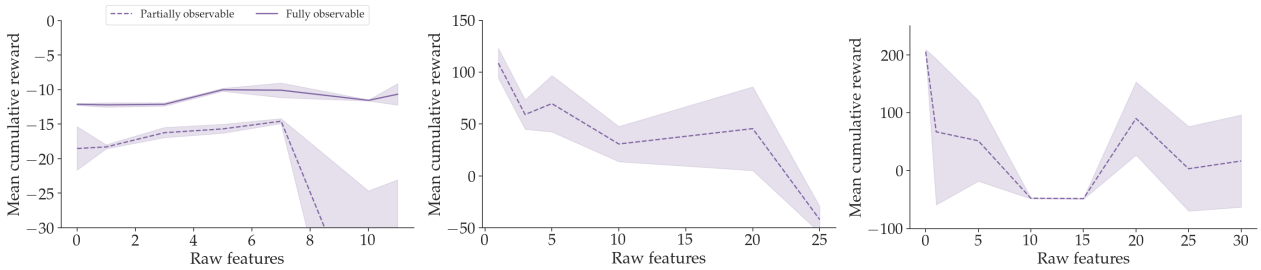
Adding Spurious Features: We next concatenate these privileged causal features with the features from the raw observation space and perform reward learning on this *augmented* feature space. Figure 2 displays a sensitivity analysis on the number of raw features. Interestingly, we find that the ability of the agent to successfully complete its task drops off as more raw features are introduced, suggesting the existence of spurious correlations. Note that the reward function still has access to the privileged features as we progressively add raw features; however, it appears that simply ignoring the concatenated raw features when learning the reward is difficult, even for a linear network, leading to causal confusion and poor policy performance.

We next inspected the weights of the learned linear reward models to understand the relative contributions of various features. Interestingly, in both the Feeding and Itch Scratching environments, the feature that corresponds to the total amount of force applied by the spoon or scratching tool is consistently given a large weighting factor. This phenomenon is likely a result of the fact that all successful Feeding and Itch Scratching demonstrations applied *some* force on the human (touching the mouth or the arm), whereas the majority of failing demos simply dropped the food or swung the end effector without applying any force on the human. This proves to be a somewhat problematic proxy; as seen in Figure 3, rather than actually feed or scratch, the robot learns the undesirable behavior of applying a large amount of force to the human's head to maximize the spoon force on human (Figure 3a) or applying a large amount of force on its own base to maximize total tool force (Figure 3b).

Discussion: Our findings in this section demonstrate the need for caution when using raw observational feature spaces originally designed for RL tasks to perform reward learning. In the three tasks we consider, the raw observation spaces introduce spurious correlations that lead to a causally-confused learned reward and poor policy performance. Using hand-selected features works well, but designing the right set of features is difficult. On top of that, we find evidence that increasing the number of features available to the reward network can exacerbate causal confusion, likely because it increases the probability that the learned reward network focuses on spurious correlations.

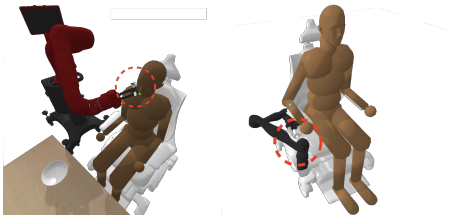
5.2. The Effect of Reward Function Model Capacity

We next study the effect of the model capacity of the reward function, r_θ. Analyzing the effect of model capacity on causal confusion is inspired by recent results in the reinforcement learning setting, where Pan et al. (Pan et al., 2022) find that increased model capacity for the policy network often increases the likelihood of reward hacking behavior. In contrast, we study the model capacity of the reward-learning network and its effect on performance.



(a) Reacher, Cumulative Reward (b) Feeding, Cumulative Reward (c) Itch Scratching, Cumulative Reward

Figure 2. **Sensitivity of preference learning to raw features in the reward network's observation.** For each environment, raw features are progressively concatenated onto the entire set of privileged features. For details on each environment's raw features, see Appendix D.

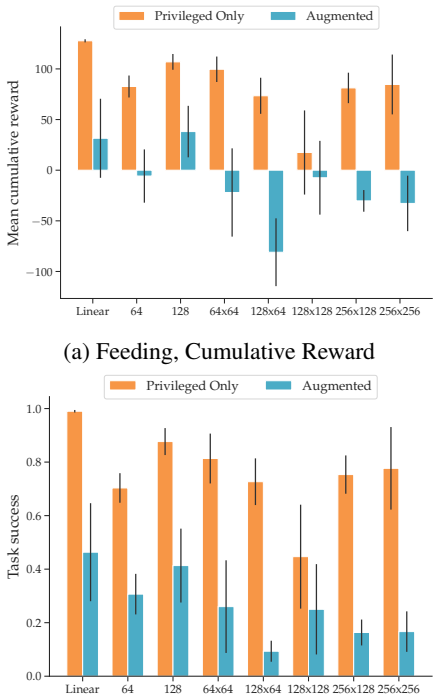


(a) Feeding (b) Itch Scratching

Figure 3. **Visualizing the spurious force feature.** Figure 3a (Feeding): The amount of force applied on the human is given a high weight, which leads to the robot applying force to various parts of the human (torso, head, etc.) rather than properly feeding. Figure 3b (Itch Scratching): The total amount of force applied by the scratching tool is given a high weight, which results in the robot preferring positions where large amounts of force are registered by the tool—here, the tool is pressed against its own base.

Increasing Model Capacity: See Appendix E for training details. We find that as the model capacity of the reward-learning network increases, the mean task success rate and cumulative ground truth reward the RL agent receives roughly decreases, all else being equal. Figure 4 shows agent performance in each environment as a function of reward-learning model capacity, trained with observations consisting of either purely privileged features or a mixture of raw and privileged features.

Discussion: As expected, the models trained on just the privileged features perform better than those trained on the augmented features, but, notably, the larger models generally perform worse than the smaller models, especially in the augmented feature space, where raw features are present. These findings go against the common belief of increasing model size to enable better generalization that is widely held in other machine learning domains (He et al., 2016; Devlin et al., 2018; Brown et al., 2020b). Our results instead suggest that the increased expressivity appears to increase the potential for inferring the wrong reward function, which the RL agent is then able to hack to a greater degree.



(a) Feeding, Cumulative Reward

(b) Feeding, Task Success

Figure 4. **Performance of preference learning as a function of reward network capacity.** *Privileged Only* denotes training the reward network on just the privileged features defined in Table 2, and *Augmented* denotes training the reward network on the privileged features augmented with 10 raw features.

6. Conclusion

We provide the first systematic study of factors that may lead to causal confusion for reward learning. In particular, we find that spurious features and increased model capacity lead to causal confusion over the true reward function, even when learning from thousands of pairwise preferences. We hope that our empirical study will inspire and facilitate future work on learning reward functions that are robust to causal confusion, even when using high-capacity neural network reward function approximators.

References

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pp. 783–792. PMLR, 2019.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on robot learning*, pp. 330–359. PMLR, 2020a.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020b.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *NIPS*, 2017.

De Haan, P., Jayaraman, D., and Levine, S. Causal confusion in imitation learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Erickson, Z., Gangaram, V., Kapusta, A., Liu, C. K., and Kemp, C. C. Assistive gym: A physics simulation framework for assistive robotics. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10169–10176. IEEE, 2020.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. *arXiv preprint arXiv:1811.06521*, 2018.

Javed, Z., Brown, D. S., Sharma, S., Zhu, J., Balakrishna, A., Petrik, M., Dragan, A., and Goldberg, K. Policy gradient bayesian robust optimization for imitation learning. In *International Conference on Machine Learning*, pp. 4785–4796. PMLR, 2021.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Pan, A., Bhatia, K., and Steinhardt, J. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Sadigh, D., Dragan, A. D., Sastry, S., and Seshia, S. A. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, Z. S. Causal imitation learning under temporally correlated noise. *arXiv preprint arXiv:2202.01312*, 2022.

Wirth, C., Akrour, R., Neumann, G., Fürnkranz, J., et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18(136): 1–46, 2017.

Zhang, J., Kumor, D., and Bareinboim, E. Causal imitation learning with unobserved confounders. *Advances in neural information processing systems*, 33:12263–12274, 2020.

A. Why study offline preference-based reward learning?

We note that, for the sake of systematically studying causal confusion, our focus in this paper is on learning reward functions from offline sets of trajectory preferences. We do this for three reasons. First, using offline trajectory preference data facilitates reproducibility by allowing us to create benchmarks (with fixed training datasets) for studying causal confusion in reward learning, similar to existing standard supervised learning benchmarks. Second, learning from offline data is known to often lead to overfitting in batch or offline reinforcement learning (Levine et al., 2020), and prior work has shown anecdotal evidence that learning a reward function from a fixed set of preferences can lead to causal confusion (Christiano et al., 2017; Ibarz et al., 2018; Javed et al., 2021)—hence, learning from offline data naturally lends itself to a study of causal confusion. Third, learning reward functions from offline data, as described above, removes the need for running reinforcement learning (Christiano et al., 2017) or trajectory optimization (Sadigh et al., 2017) in the inner loop, as is required for many active preference learning methods. This enables us to learn a reward function by optimizing the likelihood in Equation 2 (which amounts to optimizing a standard cross-entropy loss with the predicted cumulative rewards as logits) and then learn a corresponding policy using any off-the-shelf reinforcement learning algorithm to optimize r_θ. This sequential approach—learning a reward function followed by learning a policy—significantly improves the scalability and simplicity of experiments, facilitating the rapid testing and evaluation needed here for a systematic study of causal confusion.

B. Implementation and Optimization Details

In practice, we use the Adam optimizer in PyTorch to learn the reward function, r_θ, and then use PPO (Schulman et al., 2017) or SAC (Haarnoja et al., 2018) for policy optimization given r_θ.

C. Synthetic Preference Generation

To enhance scalability and reproducibility, we automatically generate a large amount of synthetic trajectory preferences. This was done using a pretrained RL policy for each of these domains that was trained using the ground-truth reward function provided with each of these environments. We then generate a large number of diverse trajectories by adding ϵ-greedy noise during policy rollouts, where ϵ is the probability that the policy takes an action uniformly at random from its action space. Thus, $\epsilon = 0$ corresponds to the fully trained RL policy and $\epsilon = 1$ corresponds to a uniformly random policy. As noted by Brown et al. (Brown et al., 2020a), adding this type of disturbance will result in monotonically decreasing performance in expectation.

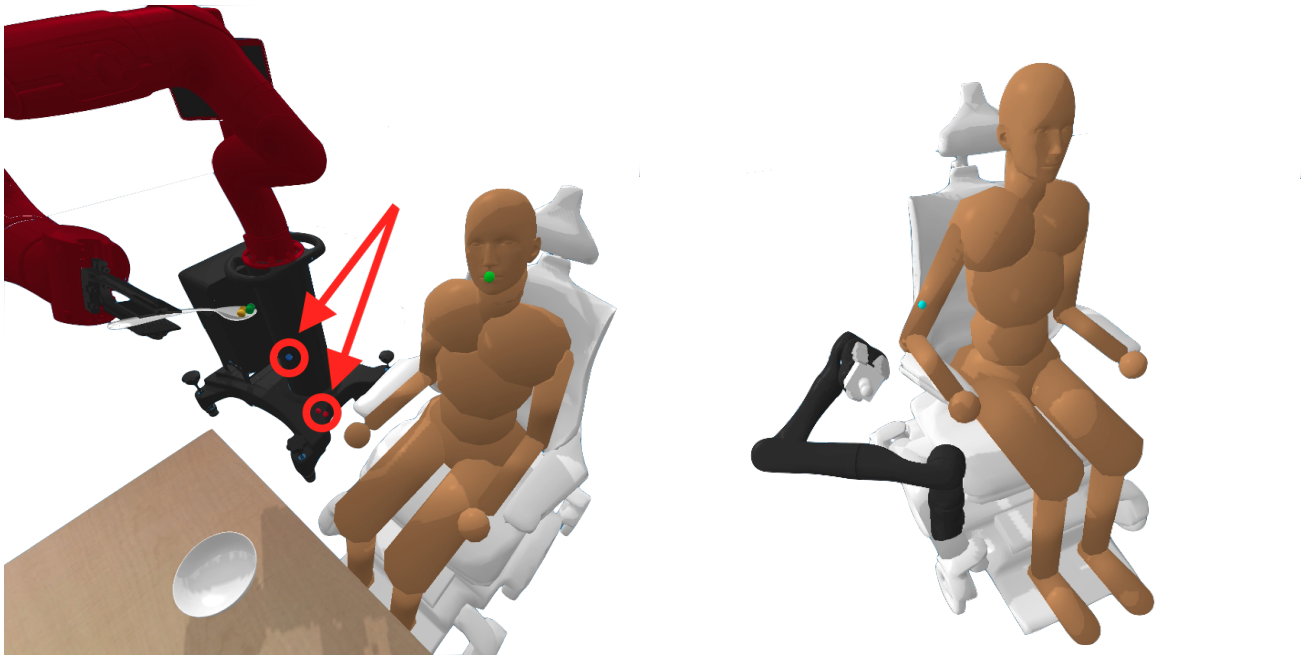
To generate pairwise preferences over trajectories, we randomly sample pairs of trajectories with replacement (while ensuring that the two trajectories within a pair are nonidentical) from a set of 120 total trajectories generated with 20 ϵ-greedy rollouts for $\epsilon \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. From these pairwise preferences, we randomly set aside 10% as a validation set, and use the rest to train. We then use the ground-truth reward functions provided by each environment to provide ground-truth preference labels. Using the dataset of preferences, we train a neural network reward function approximator with two hidden layers (128 units and 64 units, respectively) and Leaky ReLU activations, after which we perform 1,000,000 timesteps of reinforcement learning with PPO (Schulman et al., 2017) (for Feeding and Itch Scratching) and SAC (Haarnoja et al., 2018) (for Reacher) using the learned reward function in place of the ground-truth reward function. We optimize the reward function approximator using stochastic gradient descent with weight decay of 0.01 and early-stopping on the validation loss (with a patience of 10 epochs).

D. The Choice of Observation Feature Space, Experimental Details

When training, we empirically found that the model tends to learn a large bias, and that removing the bias helps incentivize RL to learn more desirable behavior. Thus, we remove biases from the final layers of subsequent models.

We train the reward-learning model on 2000 pairwise preferences of randomly-selected whole trajectories from the ϵ-greedy rollouts that are at least 60 'ranks apart' when ranked using the ground truth reward (defined as $\Delta_{pair} = 60$). We employ early-stopping with a patience of 10 epochs and maximum of 100 epochs, and apply l_1-regularization with $\lambda = 0.1$. PPO is then used to learn a policy from the learned reward for Feeding and Itch Scratching, whereas SAC is used for Reacher.

Raw features in Feeding consist of spoon and head position, robot joint angles, and amount of force applied on human, for a total raw observation dimensionality of 25. Raw features in Reacher are end effector and target position, arm angles, and angular velocities, for a total dimensionality of 11. Itch Scratching's raw features consist of tool and target position, robot



(a) Feeding (b) Itch Scratching

Figure 5. **Undesirable behavior resulting from causal confusion.** Figure 5a: The robot learns to spill the food particles (highlighted in red) in front of the human. Figure 5b: The robot learns to hover its end-effector around without scratching the itch target.

and human joint angles, and force applied by tool, for a total dimensionality of 30.

E. The Effect of Reward Function Model Capacity, Experimental Details

Similar to the above feature-space sensitivity analysis, we learn a reward function from 2000 ground-truth pairwise preference labels over randomly selected trajectory pairs with a Δ_{pair} of 60. The model is optimized using stochastic gradient descent with weight decay ($\lambda = 0.01$) and regularized with l_1-regularization ($\lambda_1 = 0.01$). We use early-stopping with a patience of 10 epochs and an upper limit of 100 epochs.

In order to measure the effect of reward network model capacity, we train the following models with increasing order of complexity: linear model, one-layer fully-connected neural network with a hidden dimension of 64, 128; two-layer fully-connected networks with hidden dimensions of (64, 64), (128, 64), (128, 128), (256, 128) and (256, 256). For each model architecture, we train the reward network on pairwise preferences, run the RL optimization for 3 seeds, and evaluate the cumulative ground truth reward and task success on 100 rollouts from a validation seed.

F. The Effect of Data Collection

Further, we explore the process through which we generate training data and the differences in agent performance that can result. Specifically, we study (1) how to generate trajectories for demonstrations, and (2) how to select or sample pairwise preferences from these generated trajectories.

Trajectory Generation: To study the effect of training data on causal confusion for preference-based reward learning, we examine two trajectory generation methods:

1. ***RL+noise.*** We first train an RL policy on a given ground truth reward, then generate demonstrations by adding ϵ-greedy noise to rollouts from the trained policy. Demonstrations are labeled using the ground truth reward.

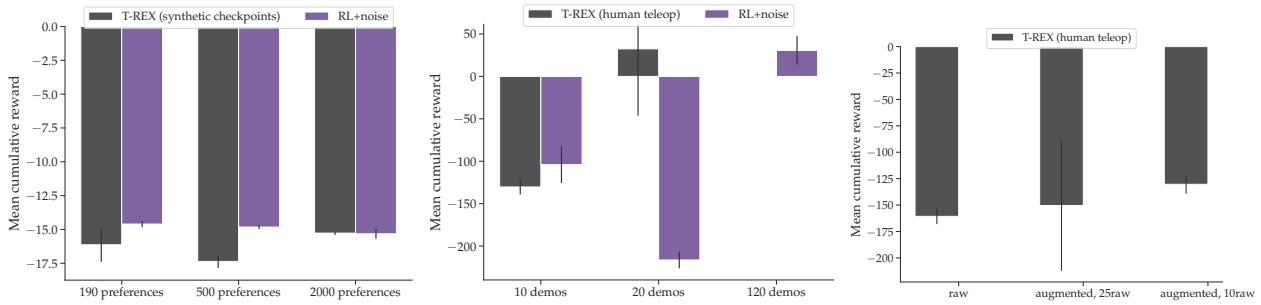
2. ***T-REX.*** Following Brown et al. (Brown et al., 2019), we also create a diverse set of training trajectories by (1) periodically checkpointing an RL policy trained on the ground-truth reward to generate *synthetic* demonstrations with varying levels of performance and (2) using *human-teleoperation* to pedagogically provide a set of trajectories with a

wide range of demonstration qualities.

Pairwise Preference Selection: Given a set of ranked trajectories, we study two ways of creating pairwise preferences:

1. ***Random Δ_{pair}-sampling.*** We randomly sample pairs of trajectories with replacement and discard pairs that are within Δ_{pair} ranks of each other. We do this until we have the desired number of pairwise preferences. We hypothesize that enforcing a Δ_{pair} difference within pairs ensures that the differences between the two trajectories are more salient and thus easier for the reward network to learn.

2. ***Systematic all-pairs selection.*** We select M evenly-spaced demonstrations from the set of provided demonstrations and generate all possible $\binom{M}{2}$ pairwise preferences. This method was originally proposed by Brown et al. (Brown et al., 2019) and requires no additional hyperparameters.



(a) Synthetic T-REX vs. RL+noise (b) Teleoperated T-REX vs. RL+noise (c) Sensitivity of demos to raw features

Figure 6. **Analysis of trajectory generation and pairwise preference selection methods.** Figure 6a: In the Reacher Environment, we use *random Δ_{pair}-sampling* with $\Delta_{pair} = 60$ to sample pairwise preferences from synthetic T-REX and RL+noise. Figure 6b: In the Feeding environment, we use *systematic all-pairs selection* to generate preferences from 20 human-teleoperated T-REX and RL+noise trajectories (the 120 demos case uses Δ_{pair}-sampling with $\Delta_{pair} = 60$). Figure 6c: We use 10 human-teleoperated demonstrations on (1) the raw features, (2) all the raw and privileged features, and (3) 10 raw and all the privileged features.

Experimental Setup: In each of the four configurations (2 generation methods $\times$ 2 preference selection methods), we train a linear model, using an augmented feature space consisting of half of the raw observation features (10 in Feeding, 5 in Reacher) and all of the privileged features (3 in Feeding, 1 in Reacher) from each environment. We train the reward learning network for 100 epochs with early stopping (patience of 10), and add weight decay with $\lambda = 0.01$ and l_1-regularization with $\lambda_1 = 0.01$. In the RL+noise trajectory generation method, we generate 20 rollouts at each level of $\epsilon = [0.0, 0.2, 0.4, 0.6, 0.8, 1.0]$. For the human-teleoperated T-REX trajectories, we had the human demonstrator perform 8 demonstrations that 'successfully complete the task', 7 demonstrations that 'fail at the task', and 5 demonstrations that 'half-succeed, half-fail' (eg., demonstration would spill some food, then feed the rest in the case of the Feeding task), for a total of 20 teleoperated demonstrations. To generate the T-REX trajectories taken from a checkpointed policy, we take 20 rollouts at each of the following checkpoints (expressed as a proportion of the total number of training iterations): $[0.01, 0.05, 0.1, 0.2, 0.8, 1.0]$. We use a fixed $\Delta_{pair} = 60$ for our random Δ_{pair}-sampling configurations. **Discussion:** Figure 6 displays our analysis of the aforementioned trajectory generation and preference selection methods when used in conjunction with one another. Firstly, we observe in Figure 6a that performing preference learning on using pairwise preferences over synthetic demonstrations generated by adding ϵ-greedy noise to a pretrained policy (RL+noise) performs on par with, if not better than, the approach suggested by Brown et al. (Brown et al., 2019) (T-REX) which takes rollouts from different partially trained policy checkpoints and then providing pairwise preferences over these rollouts. The implication is that we do not find any evidence that the causal confusion for reward inference is due to the trajectory generation being too simplistic or biased. Future work should examine whether active methods that obtain pairwise preferences by synthesizing informative trajectory pairs from scratch (Sadigh et al., 2017) lead to better performance.

Next, we note that human-teleoperated demonstrations that attempt to give a diverse set of good, mediocre, and poor demonstrations enable high preference learning performance in fewer demonstrations, as shown in Figure 6b. However, when given more demonstrations and sampled with $\Delta_{pair} = 60$, using synthetically-generated RL+noise trajectories approaches the performance of using human-teleoperated demonstrations. This reinforces the previous finding—at least for

large data regimes, the trajectory generation method is not the culprit of the poor reward inference performance demonstrated in the main results.

Lastly, we observe that human demonstrations, though preferred over synthetic demonstrations, can also lead to causal confusion over the true reward function. In Figure 6c, the mean cumulative reward increases with the addition of privileged features and the removal of raw features from the input observation of the reward network.