Gymnasium for Causal Imitation Learning

Eylam Tagor
Columbia University
eylam.tagor@columbia.edu

Abstract

Imitation learning enables agents to learn from expert demonstrations without explicit reward engineering, but standard methods can fail when experts rely on unobserved signals. In this work, we extend the CausalGym framework to implement end-to-end causal imitation learning: we operationalize both single-step and sequential π-backdoor criteria by (i) parsing environment SCMs, (ii) discovering valid adjustment sets, (iii) collecting expert data via observational queries, (iv) training causal imitators under discrete or continuous actions, and (v) evaluating through interventions. We parameterize three testbeds—MNIST digits, highway, and racetrack—each embedding latent confounders such as label noise, fog, and sensor errors. Empirically, our causal imitators recover substantially more expert reward than naive baselines in strongly confounded regimes, while matching performance when bias is mild. These results validate the π-backdoor theory in practice and demonstrate the utility of our toolkit for robust imitation learning in realistic control tasks.

1 Introduction

Imitation learning has emerged as a powerful paradigm for training agents by leveraging expert demonstrations, avoiding the need to know the reward function or estimate a surrogate for it, and eliminating the danger of exploring hazardous environments while learning. However, standard behavioral cloning (BC) and inverse reinforcement learning (IRL) methods assume that the learner has access to all of the data that are perceived by the expert and influence its behavior. In many real-world settings—autonomous driving, robotics, and human-computer interaction—experts rely on latent information or sensors unavailable to the learner. When these hidden confounders bias the observed demonstrations, naive imitation can fail catastrophically, producing policies that over- or under-react in critical situations.

Causal inference offers a principled solution to this challenge by modeling not only statistical associations but also the underlying mechanisms that generate data. Recent work on single-step causal imitation learning (Zhang et al. [2020]) and its sequential extension (Kumor et al. [2021]) establishes when and how an imitator can recover expert performance despite the presence of unobserved confounders. Yet, despite these advances, a practical and easily applicable implementation that bridges these algorithms with modern deep learning and meaningful empirical experiments has been lacking.

The CausalGym initiative is a framework that is in prime position to connect causal imitation learning algorithms with classic reinforcement learning environments. It provides abstractions for Structural Causal Models and Pearl's Causal Hierarchy alongside causal graph utility for the familiar Gymnasium library (Towers et al. [2024]).

In this paper, we build on the CausalGym foundation to deliver a complete API for performing causal imitation learning:

Preprint. Under review.

(i) **Operationalizing theory in code.** We implement the single-step and sequential π-backdoor criteria compatible with the CausalGym API, complete with adjustment set discovery, graph utilities, and policy-training modules for both discrete and continuous action spaces.

(ii) **Causal environment parameterization.** We augment various classic Gymnasium tasks with explicit SCMs that encode latent confounders, observable covariates, and expert policies, exposing rich and dynamic causal graphs.

(iii) **Empirical validation.** Through extensive experiments on these environments, we show that causal imitators outperform non-causal baselines under biasing latent conditions by correctly identifying and conditioning on valid adjustment sets.

In the following sections, we review related work (Section 2), describe our causal imitation learning framework (Section 3), detail the environment wrappers (Section 4), present experimental results (Section 5), discuss future directions (Section 6), and comment on the impact of our contributions (Section 7).

2 Literature review

We begin by situation our work within the broader field of Causal Reinforcement Learning (CRL) and its specialized task of causal imitation learning.

Causal reinforcement learning. Bareinboim et al. [2025] and Bareinboim [2025] introduce the CRL paradigm as a unifying framework where environments are modeled as SCMs and agents reason over Pearl's Causal Hierarchy to perform observational, interventional, and counterfactual queries. They show how classic RL tasks—online learning, off-policy evaluation, and causal identification—can be recast in this language. They also identify new tasks, including causal imitation learning under confounding, that lie beyond standard RL modalities.

Single-step causal imitation. Zhang et al. [2020] formalize the one-shot imitation problem in the presence of unobserved confounders, deriving a graphical π-backdoor criterion that determines exactly which observed covariates suffice to recover the expert's policy via behavioral cloning. When the criterion holds, a standard BC algorithm on the identified adjustment set guarantees expert performance; otherwise, data-dependent algorithms can still reach reasonable imitation performance.

Sequential causal imitation. Kumor et al. [2021] extend the backdoor criterion to multi-step trajectories, defining per-time-step admissible sets Z_i that unblock confounding at each decision stage. They prove that these sequential π-backdoor sets are both necessary and sufficient for imitability in episodic settings, and they give an efficient algorithm to recover them from the causal graph.

Causal IRL and GAIL. Ruan and Di [2022] and Ruan et al. [2023] bring causal adjustment into adversarial imitation (GAIL), showing how to augment the GAIL objective so that it remains robust when expert demonstrations suffer from unobserved confounding. Their Causal-GAIL wraps standard discriminator-actor-critic policy training within a causal IRL framework, matching occupancy measures only after blocking backdoor paths.

Partial identification in MDPs. Ruan et al. [2024] investigate imitation under both transition and reward confounding in full Markov Decision Processes (MDPs). They prove that if both dynamics and rewards are non-identifiable, no policy can guarantee expert performance; but when exactly one is identifiable, one can derive worst-case bounds and design robust imitation algorithms (CAIL-R and CAIL-T) that extend GAIL with partial-identification techniques.

Taken together, these works lay a comprehensive theoretical foundational of CRL principles through one-step and sequential imitation, and into more advanced settings of inverse reinforcement learning and partial identification. Each development is also accompanied by experiments that prove feasibility on low-dimensional, highly controlled tasks. In contrast, this paper scales these methods to richer environments, implements them in an extensible Python API, and benchmarks performance on complex, high-dimensional simulations.

3 Causal imitation learning

Our goal is to take the elegant, graph-based algorithms of Zhang et al. [2020] (single-step) and Kumor et al. [2021] (sequential) and weave them into a complete, end-to-end Python pipeline. Starting from a wrapped Gymnasium environment that includes an SCM and PCH interface, our code (i) introspects the causal graph, (ii) discovers valid adjustment sets, (iii) collects expert trajectories, (iv) trains a causal imitator, and (v) evaluates it under interventions. In this way we make causal imitation learning practical, reproducible, and broadly accessible.

3.1 Single-step causal imitation learning

We implement the one-shot π-backdoor algorithm of Zhang et al. [2020]. Its main components are:

1. **Graph construction.** We obtain the environment's causal graph adjacency matrix and parse it into a CausalGraph object, which captures directed and bidirected (confounding) edges.
2. **Adjustment set discovery.** We first compute the parent set
$$\mathrm{Pa}_\pi = \{\, V \in \mathrm{Observed} \setminus \{X, Y\} \mid V \notin \mathrm{Desc}(X)\},$$
then brute-force search over subsets of Pa_π, checking d-separation in a mutilated graph G_x until we find a valid π-backdoor set $Z \subseteq \mathrm{Pa}_\pi$.
3. **Expert data collection.** Using the PCH interface's `see()` method, we collect expert trajectories through simulations of the environment under the expert's behavioral policy to gather records $(obs, x, \cdots)$ for a specified number of episodes.
4. **Policy learning.** Conditioned on each observed $z \in Z$, we train a conditional generator-discriminator pair (GAN) to approximate the expert's distribution $P(X \mid Z)$.
5. **Evaluation.** We compare the causal imitator against baselines by two measures: $L1$ distance between expert and imitator histograms, and cumulative reward using semantics to enforce chosen actions.

3.2 Sequential causal imitation learning

We realize the sequential π-backdoor framework of Kumor et al. [2021] for multi-step episodes. Its key stages are:

1. **Graph utilities.** Given an environment with a sequential graph with an arbitrary number of timesteps, we obtain a CausalGraph object once again. For the sequential algorithm, various helper routines are implemented as well: `temporal_ordering`, `ancestral_graph`, `ch_plus`, `pa_plus`. These compute topological order, construct the induced ancestral subgraphs, and compute effective children and parents in the presence of latent variables, respectively.
2. **Sequential π-backdoor discovery.** Each piece of the Kumor et al. [2021] algorithm is implemented:
   (a) Finding $\mathbf{O}^X$, the set of observed variables O_i each mapped to its action X_i.
   (b) Computing the Markov boundary of $\mathbf{O}^X$.
   (c) Identifying the boundary actions whose children include unblocked variables.
   (d) Assembling conditioning sets Z_i from these components and the global temporal ordering.

   This yields a mapping $\{X_i \leftrightarrow Z_i\}$ that is both necessary and sufficient for imitation under confounding.
3. **Expert trajectories.** Expert trajectories are collected using the PCH interface's `see()` for a specified number of episodes each lasting for a specified maximum number of steps or until terminated. Full histories of (X_i, O_i) are generated.
4. **Policy training.** For each decision X_i, we extract all records at time i and train a deep neural network, using cross-entropy loss for environments with discrete action spaces and MSE loss for continuous action spaces. The result is an imitation policy set $\pi_i(o_{Z_i})$.

5. **Imitator rollout and evaluation.** Using the PCH interface's `do()` method, we deploy the imitator to perform in a real simulation of the environment, collecting cumulative rewards and comparing it to expert performance.

Together, these two modules bring the theoretical criteria of Zhang et al. [2020], Kumor et al. [2021] into a single, easy-to-use API: given any SCM-wrapped Gymnasium environment, one can automatically discover valid adjustment sets, train a causal imitator, and quantify its gains over naive baselines. Additionally, this paper expands on the sequential algorithm in Kumor et al. [2021] by generalizing it to any number of timesteps, and any configuration of inter-timestep causal relationships including handling of confounders that are sampled once, at every timestep, or as a Markov chain.

4 Environments

In our framework, each environment is a wrapper around a standard Gymnasium task (or in the case of MNIST digits, a standard dataset), augmented with two core capabilities:

1. **SCM interface.** As a subclass of CausalGym's SCM class, an environment can encode explicit structural equations for all relevant variables, including observed covariates, actions, rewards, latents, and confounders. It exposes adjacency matrices detailing the causal graph that the SCM is associated with. Additionally, each environment is equipped with a default behavioral policy that acts as the expert, making imitation learning more accessible.
2. **PCH interface.** A PCH-extending wrapper of the SCM environment enables performing queries following Pearl's Causal Hierarchy, using `see()` and `do(action)` for observational and interventional rollouts. This allows the same simulation to be acted upon by a mixture of expert and imitator policies, for example making it possible to deploy the imitator to take over in the middle of an expert demonstration.

Below we describe the environments we built, which grow in complexity from a toy proof-of-concept to a high-dimensional continuous control task.

4.1 MNIST digits

As a first sanity check, we re-implement the single-step MNIST digits experiment from Zhang et al. [2020] using the same parameterization and underlying mechanisms of the SCM. As seen in Fig. 1, the latent variable is a binary confounder U that flips the expert's class label with some probability. The observed variables are the expert's corrupted label X and the rendered digit image W. There is a surrogate, S, of the latent reward Y.

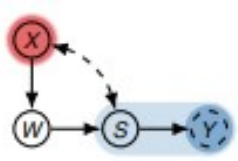


Figure 1: Causal Graph for MNIST digits.

By wrapping the MNIST digits dataset in an SCM/PCH pair, we can use the single-step algorithm from Section 3.1 to discover the π-backdoor set $\{W\}$, train a causal imitator to predict $P(S \mid W)$, and confirm that it recovers true labels despite label-noise confounding. This toy task verifies our pipeline and end-to-end before moving to richer control environments.

4.2 Highway

4.2.1 Single-step highway

We implement a one-shot imitation on the classic Gymnasium highway environment (Leurent [2018]), drawing data from the environment that mimics the highD dataset (Krajewski et al. [2018]) in the

scenario of a single longitudinal decision. The observed covariate is the velocity of the front car Z, and the ego velocity is the action X; the latents are its tail light indicator L and a weather condition U. The causal graph matches that of Zhang et al. [2020]'s highD experiment as seen in Fig. 2.

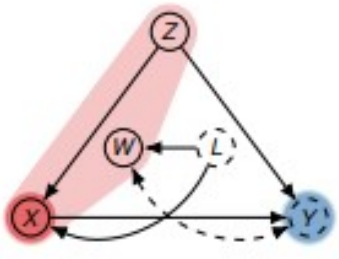


Figure 2: Causal Graph for Single-step Highway.

This environment provided a single-step representation of a Gymnasium environment, paving the way for more complex environments using sequential mechanisms, setting up for the transition to sequential imitation learning in a measurable manner.

4.2.2 Sequential highway

To move from one-shot to multi-step control, we parameterize the full Gymnasium highway environment as a time-unrolled SCM with a cumulative reward. The variables are:

- D_t: whether the vehicle in front is too close.
- L_t: ego vehicle's current lane.
- A_t, B_t: whether the left and right lanes are open, respectively.
- W_t: whether the dashboard warning sign is on.
- I_t: whether the front car's brake light is on.
- U_t: foggy weather.

The latent variable I_t, representing the front car's tail light which turns on when it is braking, is latent only to the imitator. To motivate the problem, this is because the imitator represents a drone flying far above the road, whereas the expert is the actual driver and can observe the smaller details of the environment such as I_t.

The unobserved confounder U_t, representing foggy weather (see Fig. 3a, affects: (i) the lane reading, making it noisy due to the visual impairment induced by fog; (ii) the dashboard warning light, causing it to flash with a higher probability; and (iii) the latent reward Y, which punishes reckless driving in fog as an incentive to exercise caution whereas it would otherwise reward driving fast.

(a) (b)

Figure 3: (a) Highway environment showing the unobserved confounder U_t (fog) and the yellow dashboard warning light W_t. The faint green blob below the ego vehicle signifies that due to the fog, the driver incorrectly thinks it is in the rightmost lane instead of the second-rightmost lane. Not pictured: an orange indicator for I_t at the backside of the car in front of the ego vehicle, only on when it is braking. (b) Highway environment showing what the imitator sees, in the middle of a merge toward the right lane.

This parameterization of the highway environment induces a complex SCM with many intra- and inter-timestep relationships, with substantial unobserved confounding. It also creates an issue for non-causal imitators: the observed covariate W is biased, and has no influence on the expert's decision-making. As seen in Fig. 4, there is no edge from any W_t to any X_t. In Section 5, it is demonstrated that this bias does indeed hinder performance of naive imitators, while a causal imitator can successfully ignore this bias and outperform its counterpart.

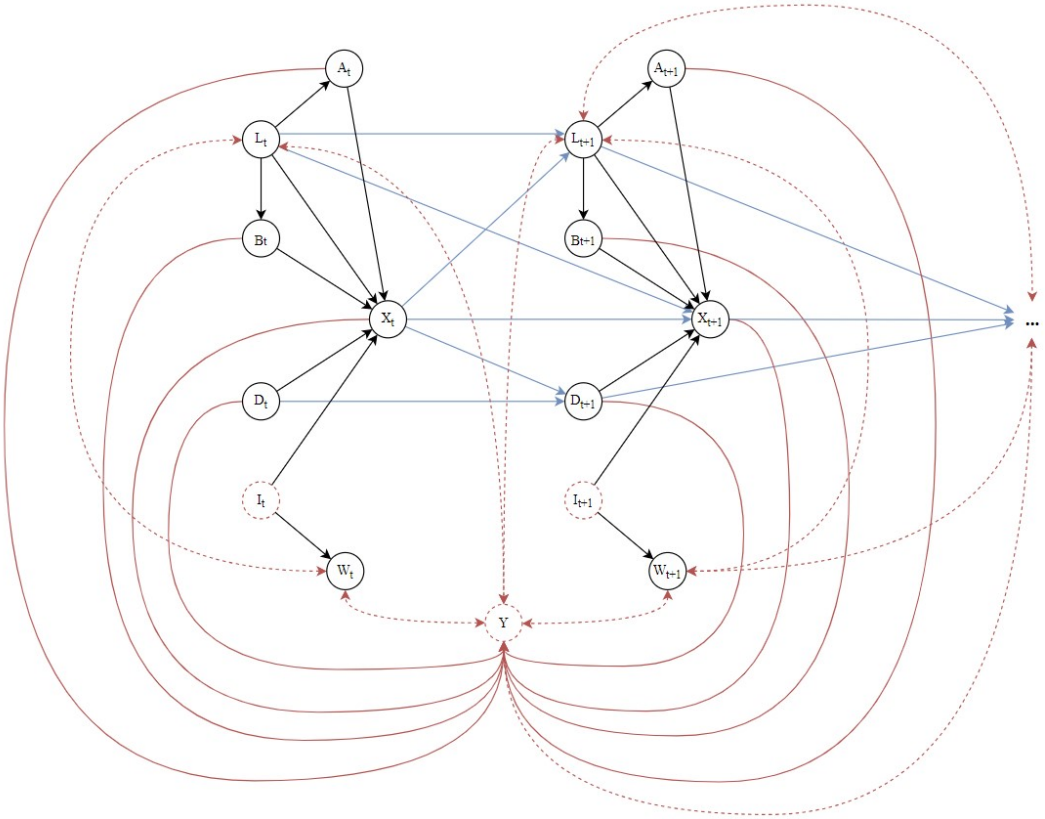


Figure 4: Causal Graph for Sequential Highway.

4.2.3 MDP highway

Although the Ruan et al. [2023] and Ruan et al. [2024] algorithms are not implemented yet, we include a variation of the highway environment that is compatible with the Causal GAIL methodology used in these algorithms for inverse reinforcement learning and partially identifiable MDPs. The principal modification from the sequential highway parameterization is the switch from a cumulative reward to one that is evaluated at every timestep, validating the Markovian assumption. See Fig. 5 for the associated causal graph.

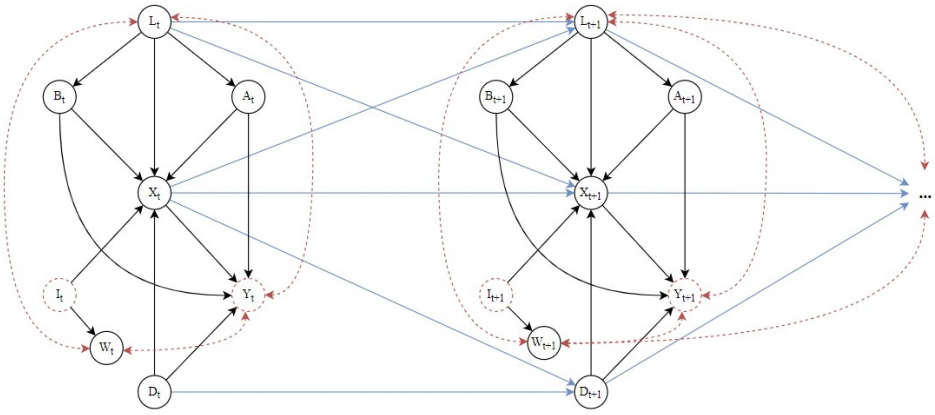


Figure 5: Causal Graph for MDP Highway.

4.3 Racetrack

We also model a sequential racetrack scenario from the same Gymnasium suite (Leurent [2018]) with two key distinctions:

1. **Continuous actions.** Instead of choosing to either speed up, slow down, merge left, merge right, or do nothing, the ego vehicle must meticulously control its heading by steering. The action space is $X_t \in [-1, 1]$. Whereas training a policy on the previous environments was achievable using a discrete input dimension and cross entropy loss, this environment necessitates functionality for handling a continuous action space and using MSE-based regression with $\tanh$ outputs.

2. **Higher-dimensional covariates.** We track a continuous lane-centering score $C_t \in [-1, 1]$ and heading error $H_t \in [-\pi, \pi]$, in addition to a binary dashboard warning light W_t to similarly introduce bias, an unobserved confounder modeling fog (U_t), and a latent covariate D—constant throughout each simulation— representing the driver being drunk or otherwise kinetically and cognitively impaired.

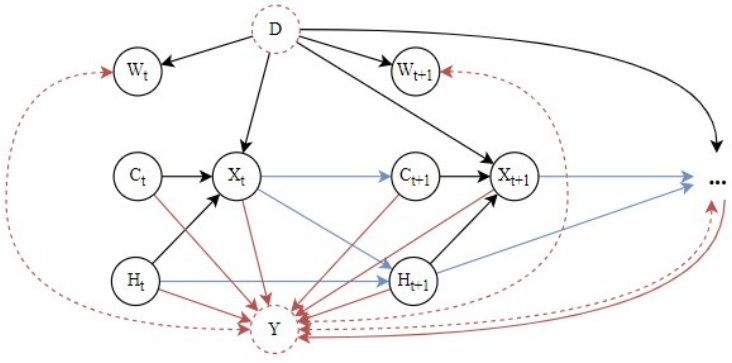


Figure 6: Causal Graph for Racetrack

The causal graph is modeled in Fig. 6. By including an environment with more realistic states and actions, we further demonstrate the practicality of causal imitation learning in real-life scenarios.

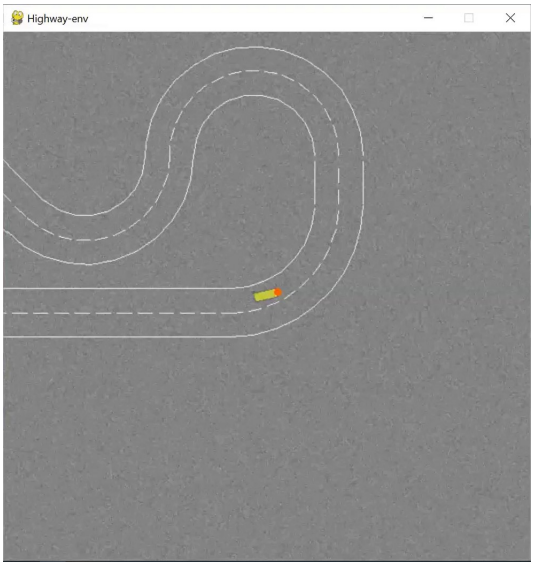


Figure 7: Racetrack environment showing U_t (foggy weather) and W_t (orange dashboard warning sign) while the ego vehicle attempts to stay at the center of the lane without slowing down.

5 Experiments

To demonstrate causal imitation learning and the feasibility of the API introduced in this paper, we focus here on the two substantive control tasks: sequential highway and racetrack.

Findings for the MNIST digits and single-step highway environments, which served as proofs of concept for the imitation learning algorithm and the CausalGym compatibility respectively, are detailed in Appendix B.

5.1 Highway

For the first significant environment, as opposed to the previous prototypes, an experiment on a much larger scale was conducted. We chose a length of 100 timesteps to fully capture temporal dependencies, encourage long-term decision making, and highlight potential growing impacts of bias over time.

Using the imitation pipeline, the sequential highway graph was unrolled into 100 timesteps and fed into the algorithm to find the set of sequential π-backdoor sets $\mathbf{Z}$ that map to each action $X_i \in \mathbf{X}$. The resulting sets included every observed covariate from $i = 0$ until current timestep $i = t$, except every $W_{0:i}$. Especially at later timesteps, this intentionally led to a high-dimensional input which serves this paper's goal of evaluating causal imitation learning under such conditions. The naive imitator's sets, however, included every observed variable as a way to represent its inability to use the causal graph to find optimal adjustment sets.

Using the default expert policy, 1000 trajectories were collected as samples for policy training. Then, a neural network was trained for each timestep to assemble a set of temporal policies for each imitator: conditioning on $Z_i \in \mathbf{Z}$ for each X_i for the causal imitator, and conditioning on $O_i \in \mathbf{O}$ for each X_i for the naive imitator.

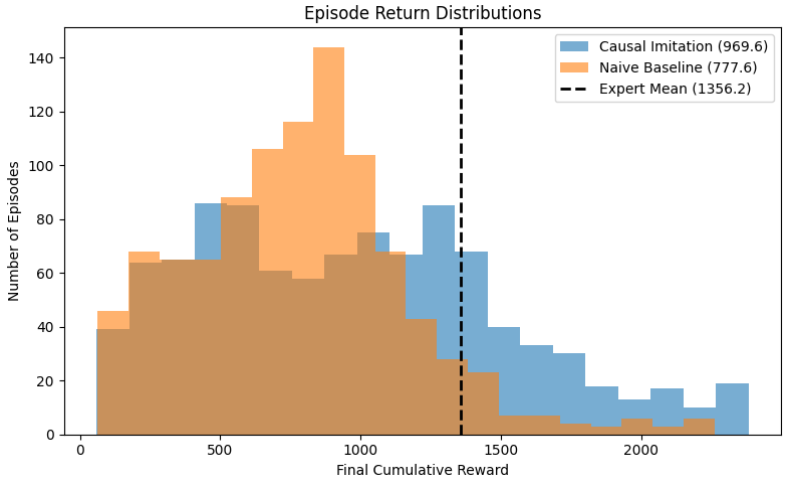


Figure 8: Expected cumulative reward per episode of at most 100 steps in the highway environment.

The policies were then rolled out for 1000 episodes each, and evaluated by comparing their reward distributions with each other as well as with the expert's. As seen in Fig. 8, although neither imitator achieves perfect imitation (likely due to the small sample size, imperfect expert, and computation limitations), it is evident that the causal imitator significantly outperforms the naive baseline.

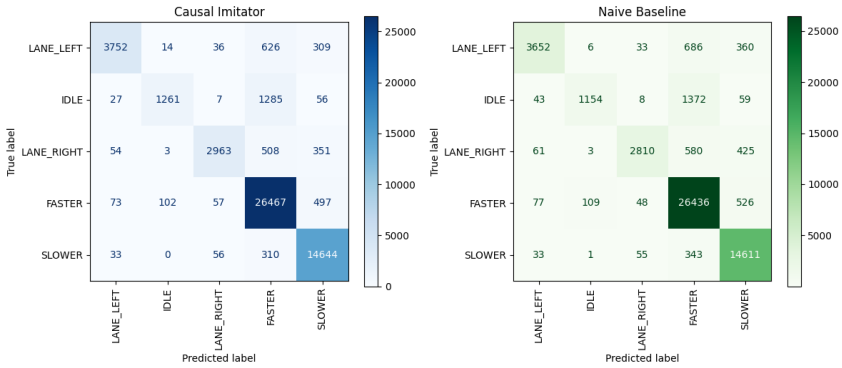


Figure 9: Confusion matrix modeling imitator action predictions before deployment in the highway environment.

In addition to measuring the imitators by average reward, we conducted an extensive analysis of the reason for this performance gap. Fig. 9 displays a confusion matrix visualizing the imitators' action prediction compared to the expert's. Applying the policies to the expert's own trajectories enabled direct comparison of the imitators, resulting in an interesting observation: the predictions are virtually the same.

This is because even the naive imitator is a deep neural network, meaning it is capable of learning its dataset to a very high accuracy. Therefore, the difference in performance would not be shown by predicting the expert's dataset, as it is still susceptible to overfitting to the biased W_t; rather, this phenomenon verifies that the difference in performance between the causal and naive imitators arose after training and during deployment, thus reinforcing the importance of the causal approach in combating safety concerns.

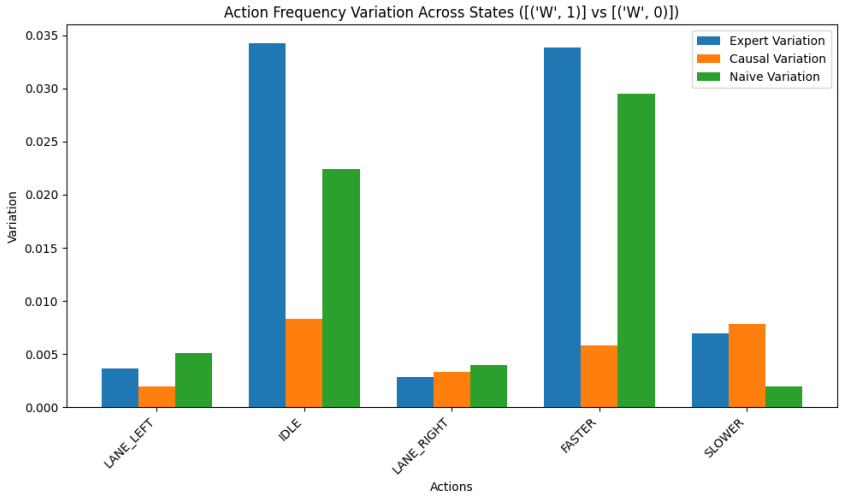


Figure 10: Action variation across states where $W_t = 0$ and $W_t = 1$ in the highway environment.

To confirm this, we compiled the trajectories from the 1000 episodes of deployment from each imitator and the expert and modeled their decision-making under different conditions, namely under different W_t values. For each agent, we split the trajectories into groups based on the observed value of W_t. Then, we calculated the difference of their action choices between the two groups, with the variation for each action is plotted in Fig. 10 (where a variation of 0 represents the exact same frequency across the two state groups for that action). The result confirms that the difference in performance can indeed be attributed to the naive imitator mistakenly assuming that the expert's policy is influenced by W_t because it notices some correlation but does not have access to the causal graph. Therefore, it overfits to the trajectories it was trained on, which in turn leads to suboptimal performance. This is corroborated by the similarity between the expert's variation and the naive imitator's variation, especially for the actions IDLE and FASTER which the expert chooses based on its estimate of U_t. In other words, the expert's variation is a result of confounding, not a causal effect; in the naive imitator's case, however, it is a causal effect. In contrast, the causal imitator does not condition on W_t at all—as seen through its near-zero variation across all actions—and ultimately achieves better imitation and better performance.

5.2 Racetrack

For the racetrack environment, the methodology from the previous experiment was repeated with a continuous action space in mind instead of a discrete action space. The results, however, failed to meaningfully distinguish between causal and naive imitation in terms of either performance or variation (See Fig. 11). The most likely explanation for this is that the underlying SCM mechanism does not induce a meaningful bias in relation to W_t, which could be due to the lack of substantial effect of U_t on either W_t or Y.

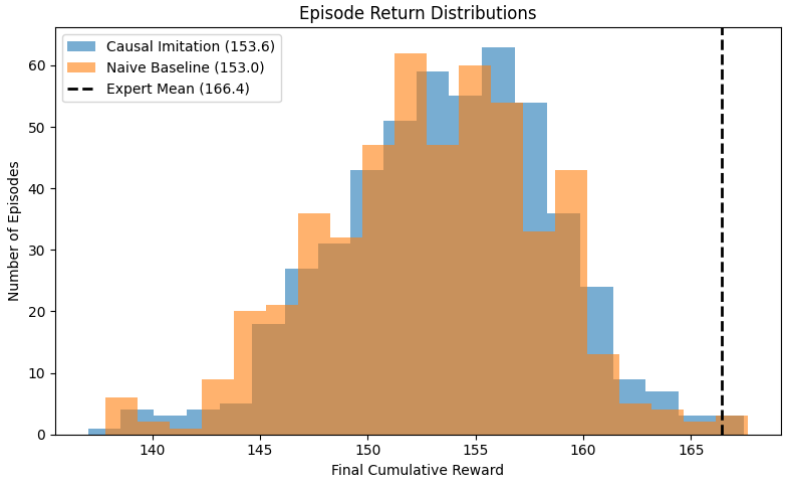


Figure 11: Expected cumulative reward per episode of 200 steps in the racetrack environment.

In summary, these experiments show that by avoiding spurious correlations, causal imitators are capable of exceeding standard behavioral cloning baselines in scenarios where bias is substantial, though they are not guaranteed to significantly outperform the baselines in every task: in lightly confounded settings, we observe only modest gains by using causal adjustment. Furthermore, it is shown that this paper's contribution, the causal imitation learning API for CausalGym, has real, empirical applicability.

6 Future Work

While our current API and experiments demonstrate the feasibility of causal imitation learning in discrete and continuous control tasks, there remain several promising directions to extend and strengthen this work:

Causal GAIL and partial identification IRL. We plan to implement the adversarial imitation and IRL algorithms of Ruan et al. [2023], Ruan and Di [2022] and Ruan et al. [2024] within the CausalGym framework. Concretely, this involves:

- Embedding the π-backdoor adjustment into the GAIL discriminator–policy loop (Causal-GAIL), so that occupancy matching occurs only over valid adjustment sets.
- Extending to partially identifiable MDPs (CAIL-R and CAIL-T), where either transitions or rewards are hidden confounded, and deriving worst-case performance bounds under partial identification.

Expanded Environment Suite. To stress-test causal imitation in richer traffic scenarios, we will wrap additional tasks from the Highway collection—such as the multi-lane merge environment and the roundabout environment—each with its own SCM parameterization and dynamic confounders (e.g. merging traffic, circulatory right-of-way). This will further validate the generality of our API and expose new forms of spurious correlation in sequential control.

Lunar Lander. The classic continuous-control Lunar Lander environment offers both high-dimensional state (position, velocity, angle, leg contact sensors) and continuous thrust controls. By constructing an SCM with latent wind gusts or sensor noise as confounders, we can evaluate causal imitation in a challenging, 2D physics domain, and compare performance against standard BC and model-based IRL approaches.

Comprehensive Benchmarking. Finally, we will conduct a large-scale empirical comparison of:

- All implemented causal imitation algorithms (single-step π-backdoor, sequential π-backdoor, Causal-GAIL, partial-ID IRL).

- A suite of environments (MNIST digits, highway variants, racetrack, merge, roundabout, lunar lander).
- Multiple confounding regimes (varying strength, time-varying vs. static, deterministic vs. stochastic confounders).

This benchmarking effort will quantify the regimes in which causal adjustment yields the greatest gains, characterize sample-efficiency trade-offs, and surface practical recommendations for practitioners.

Together, these extensions will turn CausalGym into a comprehensive experimental platform for research at the intersection of causality and reinforcement learning.

7 Conclusion

In this work, we have presented causal imitation learning functionality for CausalGym, a unified Python framework that brings modern causal inference theory into practical reinforcement learning pipelines. Starting from the π-backdoor criteria of Zhang et al. [2020] and Kumor et al. [2021], we have implemented:

- An end-to-end API for single-step and sequential causal imitation, including automated and generalized adjustment-set discovery, expert trajectory collection, and policy estimation under both discrete and continuous action spaces.
- A family of SCM/PCH wrappers for classic Gymnasium environments, each exposing rich causal graphs with latent confounders and dynamic dependencies.
- Empirical demonstrations that, in environments with substantial hidden bias, causal imitators consistent recover more of the expert's performance than standard behavioral cloning.

Beyond validating causal adjustment in high-dimensional control, our implementation lowers the barrier for researchers to explore Causal RL tasks in new domains. By integrating graph utilities, PCH queries, and flexible policy training modules, CausalGym serves both as a practical toolkit for safety-critical imitation learning.

Looking forward, we believe that extending this framework to adversarial IRL (Causal-GAIL), partial identification methods, and an expanded suite of environments will further illuminate when and how causality unlocks robust imitation. Ultimately, our goal is to make causal reinforcement learning as readily accessible and empirically grounded as the state-of-the-art deep-RL toolkits.

References

Elias Bareinboim. Causal artificial intelligence: A roadmap for building causally intelligent systems. Draft version, January 2025, 2025. URL `https://causalai-book.net/`.

Elias Bareinboim, Junzhe Zhang, and Sanghack Lee. An introduction to causal reinforcement learning. Technical report, Department of Computer Science, Columbia University
Graduate School of Data Science, Seoul National University, 2025. URL `https://causalai.net/r65.pdf`. Technical Report; first version December 2024, last version March 2025.

Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125, 2018. doi: 10.1109/ITSC.2018.8569552.

Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. Sequential causal imitation learning with unobserved confounders. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2021. URL `https://causalai.net/r76.pdf`.

Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

Kangrui Ruan and Xuan Di. Learning human driving behaviors with sequential causal imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4583–4592, 2022.

Kangrui Ruan, Junzhe Zhang, Xuan Di, and Elias Bareinboim. Causal imitation learning via inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://causalai.net/r89.pdf`.

Kangrui Ruan, Junzhe Zhang, Xuan Di, and Elias Bareinboim. Causal imitation for markov decision processes: a partial identification approach. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://causalai.net/r104.pdf`.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. Causal imitation learning with unobserved confounders. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020. URL `https://causalai.net/r66.pdf`.

A Codebase and demonstrations

The codebase of the causal imitation API and CausalGym can be found at causalgym and causalrl under the branch "eylam-imitation."

Additionally, video recordings of expert, causal imitator, and naive imitator episodes can be found at this Google Drive link.

B Proof-of-concept experiments

During the development of the causal imitation learning API and the CausalGym environment suite, the single-step MNIST digits and highway environments were used to validate the functionality of each phase of development before transitioning to the more complex sequential task.

B.1 MNIST digits

Due to the MNIST digits environment's exact mimicking of the experiment setup from Zhang et al. [2020], no large-scale experiment was conducted for this environment. Instead, we used the environment's interface to validate that the algorithm for finding π-backdoor sets in single-step scenarios is working as intended, and then, simulated a sample of expert runs (e.g. Fig. 12) to confirm that the expected reward $\mathbb{E}[Y]$ matches the previous experiment.

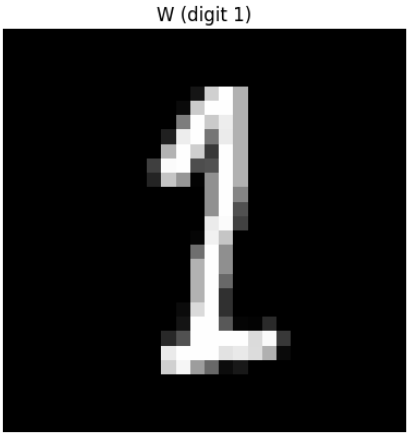
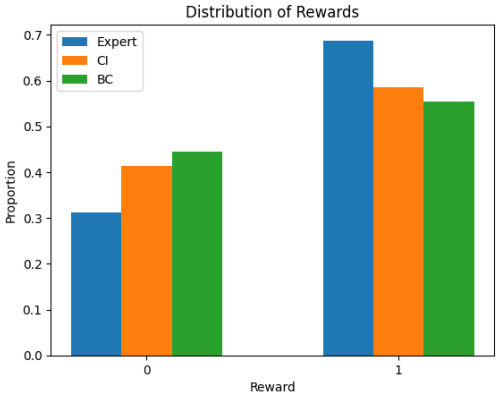


Figure 12: Example of successful digit prediction from an MNIST image.

This sanity check verified that the causal imitation pipeline is working as intended, including graph parsing, adjustment set discovery, conditional training, and PCH queries.

B.2 Single-step Highway

The objective for the single-step highway environment was to provide a transition from controlled datasets to data generation from a real stochastic environment before proceeding to sequential imitation learning. As such, the entire pipeline was invoked to train a causal imitator and a naive imitator on a collection of expert trajectories that included one state-action pair per record. Due to the small scale of this experiment and its purpose as a stepping stone toward its sequential counterparts, the results were modest in terms of performance difference between the causal and naive imitators. As seen in Fig. 13a, the causal imitator performed better by a trivial amount, although this amount has remained consistent over multiple repetitions of the experiment. The $L1$ distance showed a similar pattern of slight improvement with the causal approach, as seen in Fig. 13b.



(a)

Imitator	L_1 Distance
Causal	0.10
Naive	0.13

(b)

Figure 13: (a) Performance $\mathbb{E}[Y]$ after 128 samples in the single-step highway environment; (b) Corresponding L_1 distances between the imitators' and the expert's predictions.