CILBench: Benchmarking Robust Imitation Learning in Confounded High-Dimensional Control Environments*

Eylam Tagor
eylam.tagor@columbia.edu
Columbia University

Mingxuan Li
ml4691@columbia.edu
Columbia University

Elias Bareinboim
eb@cs.columbia.edu
Columbia University

Abstract

Imitation learning (IL) is widely used in robotics and control, yet current benchmarks assume unconfounded environments. Realistic decision-making settings routinely violate these assumptions: unobserved confounding and partial observability distort the relationship between an expert's actions and the imitator's observations, causing standard IL algorithms to overfit to spurious correlations and fail to generalize. Although recent work in causal imitation learning (CIL) provides advancements in addressing this challenge, these methods have only been evaluated in short-horizon, low-dimensional domains. We propose CILBench, a benchmark assessing IL under unobserved confounding and partial observability in high-dimensional, long-horizon control tasks modified from OGBench. We further develop methodology and an accompanying API for extracting adjustment sets in long-horizon environments, enabling scalable CIL algorithms. We observe that CIL methods consistently outperform baselines across tasks, highlighting the necessity of causal reasoning for robust imitation in complex control settings.

1 Introduction

Imitation learning (IL) has become a central paradigm in robotics and control, offering a practical alternative to reinforcement learning (RL) in domains where online exploration is costly, unsafe, or difficult to reward-engineer. Rather than optimizing a task-specific reward, an IL agent seeks to reproduce the behavior of an expert given demonstrations, which are typically collected as state–action trajectories from a policy deployed in the environment. This framework underlies a large body of work in behavioral cloning, dataset aggregation, and inverse reinforcement learning (Ross et al., 2011; Ziebart et al., 2008; Ho & Ermon, 2016; Fu et al., 2018), and is widely used in offline RL pipelines and robotics applications (Levine et al., 2020; Fu et al., 2020).

A critical assumption underlying most IL algorithms and benchmarks is that expert and imitator operate in an unconfounded environment. Concretely, the expert's action at each time step is assumed to be a function of the same observed state that will later be presented to the imitator. Under this No Unobserved Confounders assumption (NUC), the expert's policy is identifiable from observational data, and standard supervised or adversarial IL procedures can recover it given sufficient demonstrations. However, realistic decision-making systems rarely satisfy NUC. In practice, expert controllers often have access to additional sensors, internal states, or context not exposed to the imitator; physical conditions such as wind, friction, or payload can vary across time and episodes; and sensing pipelines can be partially degraded or corrupted. All of these phenomena induce unobserved confounding: latent variables that jointly affect the observed state and the expert's actions, but are not available to the imitator.

*__Work in progress.__ This v1 release includes only the AntMaze instantiation of CILBench. Additional environments and results will be included in v2.

In the presence of such confounding, naive IL methods that treat logged state–action pairs as if they were generated by a fully observed Markov decision process will in general fail to recover the expert's behavior. The situation is further complicated by the fact that the logged state in the dataset need not coincide with the information set on which the expert actually conditions. In many realistic settings, the expert's policy is a function of a rich state vector while the imitator observes only a strict subset or a corrupted projection of it, making partial observability a source of confounding in IL. From the imitator's perspective, the demonstrations then encode a mixture of effects: the expert's response to latent confounders and expert-only state components, partially aliased through the observed variables. Training on these state–action pairs without accounting for the mismatch in observation spaces leads the imitator to overfit to spurious correlations between the observed features and the expert's actions. The resulting policies may perform well on the demonstration distribution but generalize poorly under new realizations of the confounder or in deployment environments that differ slightly from the training conditions. This issue becomes especially pronounced in long-horizon, high-dimensional control tasks, where compounding errors and covariate shift can significantly degrade performance.

1.1 Causal Imitation Learning and Its Limitations

Recent work in causal imitation learning (CIL) formalizes and addresses the challenge of IL under unobserved confounding using causal graphs and structural causal models (SCMs). Zhang et al. (2020) introduces a non-sequential setting in which the expert's behavior is modeled via an SCM with latent confounders, and derives the graphical *π-backdoor criterion* to characterize when the expert's policy is imitable, or identifiable from observational data. Kumor et al. (2021) extend this framework to sequential decision-making, introducing the *sequential π-backdoor criterion* in causal MDPs for sequential imitability and providing algorithms that construct per-step adjustment sets that deconfound the expert's actions. Building on these ideas, Ruan et al. (2023; 2024) develop CIL methods based on inverse reinforcement learning and partial identification, respectively, and demonstrate the possibility of an imitator surpassing expert performance.

Despite this progress, existing evaluations of CIL remain largely restricted to low-dimensional, short-horizon settings: small discrete MDPs, low-dimensional continuous systems, or simplified driving scenarios. In these domains, the causal graphs are small, the time horizon is short, and the adjustment sets implied by the sequential π-backdoor criterion remain sensible. By contrast, modern continuous-control benchmarks, including those based on MuJoCo and related physics engines, feature state spaces with tens to hundreds of dimensions, continuous action spaces with many degrees of freedom, and horizons on the order of hundreds or thousands of steps. In such settings, naive application of existing CIL algorithms leads to adjustment sets whose size grows with the horizon, making both estimation and representation learning intractable.

At the same time, there is a growing recognition that unobserved confounding and partial observability are not edge cases but the norm in real-world robotics and control. Experts may rely on closed-loop internal states that are not logged; some sensors may fail or be removed during deployment; and external conditions such as friction, load, or disturbances may fluctuate across episodes. These factors motivate the need for CIL methods that scale to high-dimensional, long-horizon environments, and for benchmarks that reflect these practical challenges.

1.2 Benchmarks for Imitation Learning Under Confounding

Parallel to advances in algorithms, recent years have seen substantial effort devoted to building standardized benchmarks for offline and goal-conditioned RL, such as D4RL (Fu et al., 2020), OGBench (Park et al., 2025), and related suites (Towers et al., 2024; Todorov et al., 2012; Tassa et al., 2018; Yu et al., 2019). These benchmarks provide high-quality datasets and well-defined tasks, but their underlying environments are Markov and unconfounded. Consequently, empirical evaluations of CIL have remained disconnected from the broader ecosystem of RL benchmarks, and the commu-

nity lacks a standardized tool to assess whether causal methods provide tangible benefits in complex, high-dimensional environments.

1.3 Contributions: CILBench

In this work, we introduce CILBench, a benchmark for imitation learning under unobserved confounding and partial observability in high-dimensional, long-horizon control tasks. CILBench is built by systematically augmenting MuJoCo-based environments from OGBench (Park et al., 2025) with latent confounders and partial observability, and by providing a corresponding causal modeling and algorithmic framework. The key components are:

- **Confounded continuous-control environments.** For each chosen OGBench task (e.g., antmaze, humanoid maze, manipulation domains), we construct an SCM that augments the original environment with latent variables (e.g., wind fields, dynamics perturbations) that influence both the system dynamics and the expert's policy. We introduce partial observability by hiding selected state components or replacing them with noisy aggregate sensors, thereby inducing unobserved confounding between the expert's actions and the imitator's observations.

- **Scalable CIL for long-horizon tasks.** We develop a methodology and API for computing and exploiting sequential π-backdoor adjustment sets that is scalable to long-horizon environments. Our approach combines: (i) causal graph extraction from the environment, (ii) computation of base adjustment sets on short-horizon proxy graphs, and (iii) a windowed trimming and encoding scheme that yields tractable, low-dimensional representations for each time step. This pipeline enables both behavioral cloning and adversarial IL methods to incorporate causal adjustment in high-dimensional settings.

- **Expert construction.** We provide a standardized procedure to construct high-performing experts for each confounded environment by combining offline behavioral cloning on OGBench datasets with online TD3 fine-tuning under the full confounded dynamics. These experts are then used to generate demonstrations in the confounded environment, which serve as input to both causal and non-causal IL algorithms.

- **Empirical study.** We evaluate Causal BC and Causal GAIL, instantiated with our long-horizon adjustment pipeline, against naive BC and naive GAIL baselines that ignore confounding and condition on all endogenous variables. Across the suite of tasks, we observe that naive IL often fails to reach navigation goals or to complete manipulation tasks, while causal methods substantially improve performance and robustness. We further analyze the impact of expert data quantity and confounder strength, and provide qualitative visualizations of the resulting policies.

- **Extensible open-source infrastructure.** CILBench is implemented as a set of SCM/PCH environment wrappers and training utilities that integrate with existing RL libraries. The code is a precursor to **CausalGym**, a more general framework enabling researchers to easily define new confounded environments and to assess causal algorithms for a broader range of RL tasks.

By bridging the theory of causal imitation learning with the practical realities of high-dimensional continuous control, CILBench provides a benchmark that systematically evaluates IL algorithms under unobserved confounding in realistic environments. We hope that this work will serve both as a testbed for future CIL methods and as a step toward deploying robust imitation learning systems in real-world, confounded settings.

2 Causal Imitation Learning

We model the joint expert–environment system as a structural causal model $M = \langle \mathbf{U}, \mathbf{V}, \mathscr{F}, P(\mathbf{u}) \rangle$, where $\mathbf{U}$ is a set of exogenous (noise) variables, $\mathbf{V}$ is a set of endogenous variables, $\mathscr{F} = \{f_V : V \in \mathbf{V}\}$ are structural assignments, $V \leftarrow f_V(\text{pa}(V), U_V), \forall V \in \mathbf{V}$, and $P(\mathbf{u})$ is a distribution over $\mathbf{U}$. $\mathbf{X} \subseteq \mathbf{V}$ is the action set, and $Y \in \mathbf{U}$ is the latent reward. The induced causal diagram G has one node for each $V \in \mathbf{V}$, directed edges from $\text{pa}(V)$ into V, and bidirected edges between variables with a shared unobserved parent. In our setting, $\mathbf{V}$ includes states, actions, rewards, and latent environment variables.

The imitator does not observe all of $\mathbf{V}$. We partition the endogenous variables into

$$\mathbf{V}^O \subseteq \mathbf{V} \quad \text{(observed to the imitator)}, \qquad \mathbf{V}^L = \mathbf{V} \setminus \mathbf{V}^O \quad \text{(latent to the imitator)}.$$

The expert demonstrations reflect the joint observational distribution $P(\mathbf{V}^O)$, whereas the imitator, operating under its own policy, induces the interventional distribution $P(\mathbf{V} \mid do(\pi))$. In the presence of latent variables $\mathbf{V}^L$, these two distributions may differ substantially: correlations between observed variables and expert actions may be driven by unobserved confounders rather than causal structure. The role of causal imitation learning is therefore to determine, for each time step t, which subset of observed variables $\mathbf{Z}_t \subseteq \mathbf{V}^O$ is sufficient for constructing an unbiased approximation of the expert's decision mechanism, $\pi_t(\mathbf{x}_t \mid \mathbf{Z}_t) \approx P(\mathbf{X}_t \mid \mathbf{Z}_t)$, in a way that is stable to the removal of latent confounding.

To formalize this requirement, the sequential π-backdoor criterion graphically characterizes what each $\mathbf{Z}_t$ must contain so that conditioning on $\mathbf{Z}_t$ blocks all spurious (noncausal) paths from $\mathbf{X}_t$ to the final outcome Y.

Definition 1 (Sequential π-Backdoor Criterion (Kumor et al., 2021))**.** Let G be the causal diagram induced by the SCM. For each action $\mathbf{X}_t$, define a manipulated graph $G_t^{'}$ obtained by: (i) removing all incoming edges into future actions $\mathbf{X}_{t+1:H}$, and (ii) replacing each future action $\mathbf{X}_j$ $(j > t)$ by a node whose parents are restricted to $\mathbf{Z}_j$. A family of sets $\{\mathbf{Z}_t\}_{t=0}^{H}$ satisfies the sequential π-backdoor for $(G, \mathbf{X}, Y)$ if, for every t, $(\mathbf{X}_t \perp\!\!\!\perp Y \mid \mathbf{Z}_t)_{(G_t^{'})_{X_t}}$ or $\mathbf{X}_t \notin \text{An}_{G_t^{'}}(Y)$. Here $(G_t^{'})_{\mathbf{X}_t}$ denotes the graph obtained from $G_t^{'}$ by deleting outgoing edges from $\mathbf{X}_t$.

When $\{\mathbf{Z}_t\}$ satisfies Definition 1, conditioning on $\mathbf{Z}_t$ removes all confounding and noncausal dependencies between $\mathbf{X}_t$ and Y that arise from shared latent parents in $\mathbf{V}^L$, proxy variables, or unobserved factors influencing both the expert's action and future state transitions. Crucially, $\mathbf{Z}_t$ is restricted to the imitator's observation set $\mathbf{V}^O$. If some component of the true backdoor set lies in $\mathbf{V}^L$, it becomes an unobserved confounder and the imitator cannot recover the expert's conditional policy $P(\mathbf{X}_t \mid \mathbf{Z}_t)$ without bias. In such cases, naive behavioral cloning on all available observations incorrectly conditions on variables that violate the backdoor separation, amplifying confounding rather than removing it and ultimately failing to imitate. Learning a conditional policy $P(\mathbf{X}_t \mid \mathbf{Z}_t^O)$ where $\mathbf{Z}_t^O \subset \mathbf{V}^O$ and $\mathbf{Z}_t^O \subset \mathbf{Z}_t$, however, can approximate the expert policy despite the broken imitability condition.

The sequential π-backdoor criterion therefore provides the formal grounding for the adjustment sets used throughout CILBench. These sets determine which components of the observed state history are safe and necessary to condition on, and which should be excluded to avoid encoding spurious dependencies introduced by latent confounders. Due to the impracticality of the algorithm for environments of CILBench's size, we approximate the criterion by exploiting structural properties of the environments as seen in Appendix A.

2.1 Example Confounded Sequential Graph

To ground the discussion, we consider the high-level sequential causal diagram in Fig. 1 that is a generalization of CILBench environments. S_t are endogenous state variables, X_t are actions, W_t

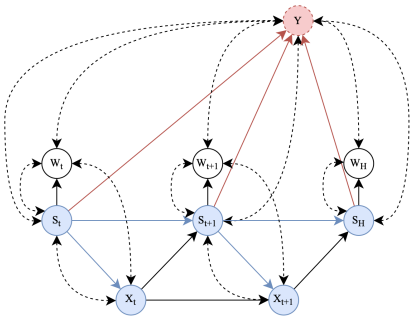


Figure 1: High-level causal diagram for a confounded sequential control setting. States S_t evolve under actions X_t, while latent variables jointly influence the evolution of S, the decision of X, the spurious measurements W, and the outcome Y. The imitator observes $\{S, W\}$, while the expert may condition on additional state components represented by bidirected edges to X.

are endogenous state variables that are spurious or noisy measurements, and Y is the terminal reward (e.g., success/failure). Solid arrows encode causal influences while bidirected edges encode unobserved confounding through exogenous state variables.

In this diagram, the expert's policy at time t is a function of S_t and the unobserved confounders shown through bidirected edges into X_t. The terminal outcome Y depends on the entire trajectory through $S_{t+1:H}$ and latent influences.

From the perspective of the sequential π-backdoor, valid adjustment sets $\mathbf{Z}_t$ for X_t must not condition on the spurious W_t when these act as proxies for latent confounders or colliders. In particular, if W_t is a noisy function of an unobserved disturbance that also affects transitions and reward, conditioning on W_t may create spurious associations between X_t and Y along paths that are blocked in the interventional regime. For this reason, in CILBench environments the correct backdoor sets for actions must be built from subsets of the physical state S alone, and not include the W-nodes.

3 Environment Design

CILBench is constructed by taking existing continuous-control tasks from OGBench, wrapping them in SCMs $\langle \mathbf{U}, \mathbf{V}, \mathscr{F}, P(\mathbf{u}) \rangle$ that induce latent confounding and partial observability, and providing a Pearl Causal Hierarchy (PCH) interface (Bareinboim et al., 2022) that exposes observational, interventional, and counterfactual modes required by causal imitation algorithms.

For example, consider the AntMaze environment. We parameterize the underlying MuJoCo state at time t into the following endogenous variables:

- $P_t \in \mathbb{R}^3$: torso position,
- $O_t \in \mathbb{R}^4$: torso orientation quaternion,
- $A_t \in \mathbb{R}^8$: joint angles,
- $L_t \in \mathbb{R}^3$: torso linear velocity,
- $T_t \in \mathbb{R}^3$: torso angular velocity,
- $J_t \in \mathbb{R}^8$: joint angular velocities,
- $X_t \in \mathbb{R}^8$: joint torque action,
- $Y_H \in \mathbb{R}$: latent terminal reward at horizon H.

For imitation learning, we make O_t observable only to the expert. To induce confounding, we introduce an exogenous two-dimensional wind field $U_t \in \mathbb{R}^2$ that evolves stochastically over time and is never observed by neither the imitator nor the expert. This wind exerts a horizontal force on the torso body at each time step, perturbing the true transition dynamics. We also define an observed but noisy heading sensor $W_t \in \mathbb{R}^2$ as a mixture of the agent's yaw-based heading (O_t) and the normalized wind direction (U_t), with added Gaussian noise and a risk of distribution shift between environment instances. Intuitively, W_t acts as a corrupted proxy that is confounded with Y_H through U_t and, to the imitator, X_t and other state variables through O_t. The SCM dynamics are implicitly given by the MuJoCo simulator and our confounder injection:

$$\begin{aligned} P_{t+1}, O_{t+1}, A_{t+1}, L_{t+1}, T_{t+1}, J_{t+1} &\leftarrow f_{\text{MuJoCo}}(P_t, O_t, A_t, L_t, T_t, J_t, X_t, U_t), \\ U_{t+1} &\leftarrow f_U(U_t, \epsilon_t^U), \\ W_t &\leftarrow f_W(O_t, U_t, \epsilon_t^W), \\ Y_H &\leftarrow f_Y(P_H, U_{0:H}), \end{aligned}$$

where f_U is a piecewise-constant gust process, f_W combines orientation and wind direction, and f_Y encodes success-at-goal with a penalty proportional to wind magnitude and distance-to-goal. The induced causal graph includes directed edges capturing the physical dependencies (e.g., $J_t \to A_t$, $L_t \to P_t$, $X_t \to J_{t+1}$) and bidirected edges representing unobserved common causes, such as between W_t and Y_H via U_t, as detailed in Appendix B.

3.1 Expert Demonstrations

Expert policies in CILBench are constructed using offline-to-online RL detailed in Algorithm 1. For each environment, we begin by training a goal-conditioned behavioral cloning policy on provided demonstrations from the base OGBench environment. This policy provides an initialization that captures the global structure of the task, but it is not yet ready for the confounders introduced in the modified environment. To obtain an expert that reflects performance under the confounded dynamics, we then fine-tune this BC policy through a period of off-policy TD3-style actor–critic training. Reward shaping is added optionally during fine-tuning to compensate for the sparse reward signals in long-horizon tasks. The result of this stage is a strong expert policy capable of operating effectively under the latent disturbances, partial observability, and altered transition dynamics of the confounded environment. Collecting expert demonstrations is done through the environment's PCH wrapper by calling `env.see()`.

Algorithm 1 Full Imitation Learning Procedure

Require: OGBench dataset $\mathcal{D}_{\text{OG}}$, confounded env $\mathcal{E}_{\text{conf}}$, lookback k.

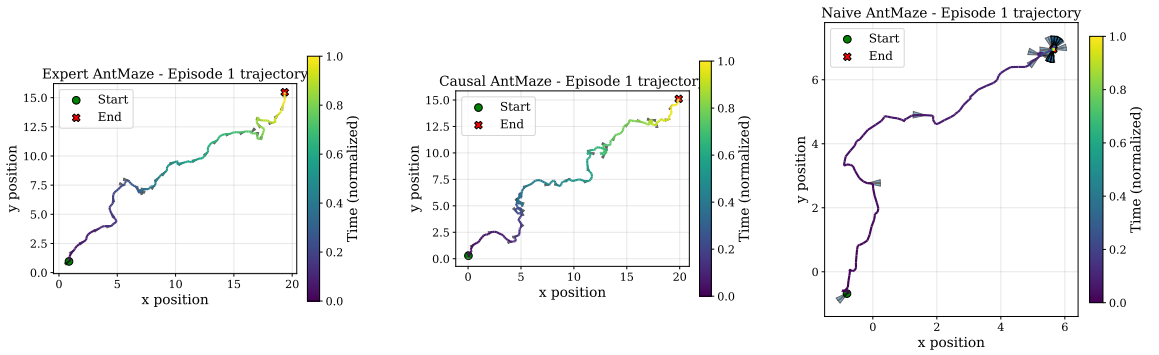
1: **Train BC expert on OGBench:** $\pi_{\text{BC}} = \arg\min_\pi \mathbb{E}_{(s,a)\sim\mathcal{D}_{\text{OG}}}\big[\ell(\pi(s), a)\big]$.
2: **TD3 fine-tuning in confounded env:** initialize replay buffer; pretrain critics offline; fine-tune actor online to obtain expert π_{exp}.
3: **Compute windowed adjustment:** apply Algorithm 2 to get $\{\mathbf{Z}_t^H\}$ and *Slots*.
4: **Collect expert trajectories in $\mathcal{E}_{\text{conf}}$:** $z_t = \text{Encode}(o_{\le t}; \mathbf{Z}_t^H, \mathit{Slots})$.
5: **If Causal BC:** $\hat{\pi}_\theta = \arg\min_\pi \mathbb{E}_{(z_t,a_t)}\big[\ell(\pi(z_t), a_t)\big]$.
6: **If Causal GAIL:** train discriminator $D_\omega(z, a)$ and actor $\pi_\theta(a \mid z)$ via

$$\max_\pi \; \mathbb{E}_\pi[\log D_\omega(z,a)] \; + \; \mathbb{E}_{\text{exp}}[\log(1 - D_\omega(z,a))],$$

with PPO/GAE updates on windowed features.
7: **return** the trained causal imitator $\hat{\pi}_{\text{CIL}}$ (BC or GAIL variant).

3.2 Evaluation Protocol

Policies are evaluated based on measures including average return $\mathbb{E}[\sum_{t=0}^{H-1} r_t]$, task-specific success rate, and summary statistics of trajectory-level outcomes (e.g., distance to goal).



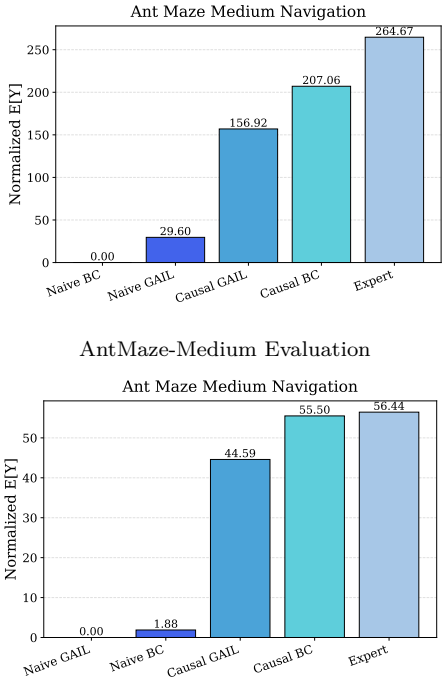
Expert | Causal Imitator | Naive Imitator

Figure 3: Position heatmaps for expert, causal, and naive policies on AntMaze. The expert and causal imitator reliably reach the goal region, while naive imitation collapses near the start.

4 Experiments

We evaluate imitation learning under confounding in two representative long-horizon tasks from CILBench: AntMaze-Medium and AntMaze-Large. For each environment, we compare four algorithms: Naive BC, Causal BC, Naive GAIL, and Causal GAIL. All models share identical architectures, optimizers, and training schedules; causal variants differ only in the conditioning sets used by their encoders. This isolates the effect of causal adjustment from architectural or algorithmic confounds.

Policies are deployed in interventional mode using the PCH wrapper (`env.do()`), ensuring that evaluation measures the performance of the learned policy rather than confounded observational statistics. We report average episode return, task-specific success rate, and qualitative trajectory summaries. Each result is averaged over multiple episodes with fixed seeds for comparability.



AntMaze-Medium Evaluation

AntMaze-Large Evaluation

Figure 2: Evaluation returns for AntMaze-Medium and AntMaze-Large. Causal variants consistently outperform naïve methods across scales. (Ignore erroneous bottom plot title.)

4.1 AntMaze Navigation

AntMaze requires navigating a quadruped robot through a maze to a distant goal over a horizon of $H = 1000$. In the confounded version, the MuJoCo dynamics are perturbed by a latent wind field, and the expert has access to privileged orientation information hidden from the imitator. The imitator instead receives a mixed proxy sensor W_t that combines orientation and wind direction, making it a confounded measurement that violates the sequential π-backdoor criterion. Expert policies are obtained following the offline-to-online pipeline described in Algorithm 1.

4.2 Results

Figure 2 summarizes quantitative results. Causal BC and Causal GAIL reliably reach the goal in AntMaze-Medium, achieving substantially higher returns and success rates than their naive counterparts, which fail to make meaningful progress beyond the start region. Trajectory heatmaps in Figure 3 confirm that naive IL encodes spurious dependencies arising from the confounded proxy sensor and stalls early in the maze, while causal variants reproduce the expert's global navigation strategy.

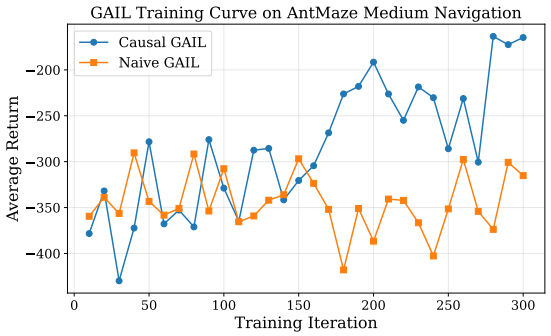


Figure 4: GAIL learning dynamics on AntMaze-Medium. Causal GAIL eventually surpasses Naive GAIL in discriminator-derived reward, corresponding to improved downstream performance.

On AntMaze-Large, where solving the maze is significantly more difficult, the same pattern persists: naive IL collapses, while causal variants recover competitive expert-level behavior. Learning curves in Figure 4 further illustrate that the discriminator in Causal GAIL yields a reward signal that supports stable policy improvement, whereas Naive GAIL's discriminator overfits to spurious proxy correlations.

Across both tasks, the magnitude of the gap between causal and naive variants demonstrates that correct adjustment of conditioning variables is the key determinant of performance in confounded long-horizon control, overshadowing differences between supervised (BC) and adversarial (GAIL) training.

5 Discussion

The experiments highlight several phenomena that are difficult to observe in standard unconfounded benchmarks but become explicit in the confounded, long-horizon settings of CILBench.

5.1 Failure Modes of Naive Conditioning

Across all settings, naive IL fails not by small margins but by converging to qualitatively incorrect behavior. This is consistent with the sequential π-backdoor theory: conditioning on proxy variables that are influenced by latent confounders opens noncausal paths between the action and outcome variables. In practice, this induces stable but incorrect action mappings that reflect spurious observational dependencies rather than the expert's decision rule. The fact that this failure persists despite expressive function approximators and adversarial training indicates that representation-level confounding, rather than lack of optimization, is the central obstacle.

5.2 Primacy of Causal Conditioning Over Algorithmic Sophistication

Once the adjustment sets are correctly specified and encoded, the performance gap between Causal BC and Causal GAIL is significantly smaller than the gap between causal and naive variants. This suggests that, under confounding, the dominant factor affecting IL performance is the choice of conditioning variables, not the choice of learning paradigm. Adversarial IL methods cannot recover the expert policy when the discriminator operates on confounded features, as it is incentivized to exploit spurious expert–imitator differences. The windowed causal representation eliminates these differences, enabling both BC and GAIL to approximate the expert policy effectively.

5.3 Designing Confounded Benchmarks with Controlled Difficulty

Constructing environments that meaningfully stress-test causal IL requires balancing two competing factors: (i) the confounders must be strong enough to falsify the NUC assumption and induce observable degradation in naive IL, yet (ii) expert behavior must remain recoverable using variables available to the imitator. CILBench's construction—based on latent forces, hidden state components, and proxy sensors—provides a controlled setting where sequential π-backdoor adjustment remains feasible, informative, and computationally tractable after windowing. This offers a template for designing future confounded benchmarks in both simulation and real-robot domains.

5.4 Connections to Causal RL and Representation Learning

The windowed adjustment sets produced in CILBench function as structured, causally justified representations. Rather than learning arbitrary embeddings, we restrict policy inputs to variables that satisfy a truncated sequential π-backdoor criterion. This connects CILBench to broader efforts in causal reinforcement learning and causal representation learning, where explicit structural restrictions are used to separate causal from spurious information. Our results suggest that such structure is essential when transitioning imitation algorithms from small synthetic domains to long-horizon continuous-control tasks.

5.5 Limitations

Our confounders are hand-designed and time-homogeneous; extending CILBench to settings with unknown or nonstationary confounders is an important direction for future work. Moreover, while the windowed adjustment procedure scales to the horizons considered here, more general causal structures may require adaptive or learned temporal scopes. Finally, our evaluation focuses on BC and GAIL; a broader study including AIRL, offline RL variants, and model-based causal methods would provide a more complete picture of how causal adjustment interacts with diverse IL algorithms.

6 Conclusion

CILBench establishes a foundation for studying imitation learning in settings where unobserved confounding and partial observability are not edge cases but intrinsic features of real decision-making systems. By enabling controlled, high-dimensional evaluations grounded in structural causal models, the benchmark creates a bridge between causal inference, continuous-control RL, and robotics, allowing these communities to investigate questions previously confined to toy domains.

The broader impact of such a platform is twofold. First, it provides a scientifically grounded way to stress-test IL algorithms under conditions resembling those encountered in real-world embodied agents whose sensors or internal states differ from those available during deployment. Second, it offers a unifying testbed for causal-representation and robust-policy research, allowing methods from adjacent fields—causal discovery, partial identification, counterfactual RL—to be evaluated in environments where latent structure is explicitly defined and manipulable.

CILBench is designed to be extensible: new confounded variants of manipulation, locomotion, or multi-agent tasks can be added simply by specifying an SCM and observation map. We envision the benchmark evolving into a shared repository of confounded environments, supporting systematic comparisons across causal and non-causal algorithms and accelerating progress toward decision-making systems that remain reliable under hidden disturbances, missing sensors, and distribution shift.

Ultimately, the benchmark aims to catalyze a shift in how imitation learning is evaluated: from assuming idealized, fully observed MDPs to rigorously accounting for latent structure and its consequences. We hope CILBench will serve as a stepping stone toward the development of robust, causally grounded agents capable of operating safely and effectively in the complex, confounded environments that characterize real-world control.

References

Elias Bareinboim, Juan D. Correa, Duligur Ibeling, and Thomas Icard. *On Pearl's Hierarchy and the Foundations of Causal Inference*, pp. 507–556. Association for Computing Machinery, New York, NY, USA, 1 edition, 2022. ISBN 9781450395861. URL https://doi.org/10.1145/3501714.3501743.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1710.11248.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. In *NeurIPS*, 2020. URL https://arxiv.org/abs/2004.07219.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *NeurIPS*, 2016. URL https://arxiv.org/abs/1606.03476.

Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. Sequential causal imitation learning with unobserved confounders. In *NeurIPS*, 2021. URL https://causalai.net/r76.pdf.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives. *arXiv preprint arXiv:2005.01643*, 2020. URL https://arxiv.org/abs/2005.01643.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://github.com/seohongpark/ogbench.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Kangrui Ruan, Junzhe Zhang, Xuan Di, and Elias Bareinboim. Causal imitation learning via inverse reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 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 *NeurIPS*, 2024. URL https://causalai.net/r104.pdf.

Yuval Tassa, Yotam Doron, Alistair Muldal, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018. URL https://arxiv.org/abs/1801.00690.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024. URL https://arxiv.org/abs/2407.17032.

Tianhe Yu, Deirdre Quillen, Chelsea Finn, et al. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1910.10897.

Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. Causal imitation learning with unobserved confounders. In *NeurIPS*, 2020. URL https://causalai.net/r66.pdf.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, 2008. URL https://www.aaai.org/Papers/AAAI/2008/AAAI08-227.pdf.

A Causal Adjustment in High-Dimensional Long-Horizon Environments

The sequential π-backdoor criterion (Def. 1) gives a graphical solution to the adjustment problem in confounded sequential decision-making. However, naively applying it to modern continuous-control tasks quickly becomes intractable. A single episode in our antmaze setting has horizon $H = 1000$, with state dimension on the order of tens of real-valued variables. Unrolling the SCM over the full horizon yields a graph with $O(H)$ time-indexed copies of each endogenous variable (states, actions, auxiliary measurements, rewards), and potentially dense bidirected structure induced by latent confounders. In such graphs, the FINDOX procedure of Kumor et al. (2021) (which returns the maximal admissible set $\mathbf{V}_X^O$ for sequential π-backdoor adjustment) must operate over thousands of nodes, and the resulting π-backdoor sets $\{Z_t\}$ may grow linearly in H even when the underlying dynamics are Markovian. From the perspective of function approximation, this implies that a policy network at time t may need to condition on the full history of observed states and actions, which is statistically inefficient and computationally prohibitive.

Algorithm 2 Feasible Windowed Sequential π-Backdoor Adjustment

Require: Lookback k, full horizon H.

Construct proxy environment $\mathcal{E}_k$ with horizon $h = k + 1$ and extract its causal graph G_k.
Compute the observable parent map $\mathbf{O}^X$ on G_k (FINDOX).
Determine Markov boundary MB and boundary actions BA in the ancestral graph.
for $t = 0, \ldots, h-1$ **do**
  Compute $\mathbf{Z}_t^k \subseteq \mathbf{V}^O$ satisfying the sequential π-backdoor criterion on G_k.
end for
for $X_i \in \mathbf{X}$ **do**
  $Z_t = \{ (v, \tau) \mid (v, \tau - (t - (h-1))) \in Z_{h-1}^k \}$.
end for
for $t = 0, \ldots, H-1$ **do**
  $Z_t^H = \{ (v, \tau) \in Z_t \mid \tau \geq t - k \}$.
end for
Build window specification *Slots* by enumerating lags $\{-1, \ldots, -k\}$ for each observed variable $V \in \mathbf{V}^O$ with its dimension.
return $\{Z_t^H\}_{t=0}^{H-1}$, *Slots*.

To make causal adjustment feasible in this regime, we exploit two structural properties of the environments in CILBench. Firstly, the SCM is time-homogeneous, and each time slice has the same local structure: the parents and children of a variable at time t are isomorphic to those at time $t+1$, up to boundary effects at the start and end of the episode. Secondly, confounding and causal influence have a bounded temporal span: there exists a window length k such that, conditional on the last k time steps of the relevant observed variables, older history does not change the admissible adjustment set for $\mathbf{X}_t$ with respect to Y.[1] Under these assumptions, we can (i) solve the sequential π-backdoor problem on a short-horizon proxy graph, (ii) reuse the resulting backdoor structure across time by shifting indices, and (iii) trim each Z_t to a fixed lookback window of length k. We now formalize this pipeline as an algorithm that takes as input the causal graph of a confounded environment and outputs a windowed representation suitable for high-dimensional CIL.

B Expanded Causal Graphs

[1] Formally, we assume that the ancestors of $\mathbf{X}_t$ and Y that lie in the imitator's observation set can be reasonably captured within a fixed-width window of length k in the unrolled graph. In our environments, this corresponds to the Markovian nature of the MuJoCo dynamics together with confounders that do not have arbitrarily long memory.

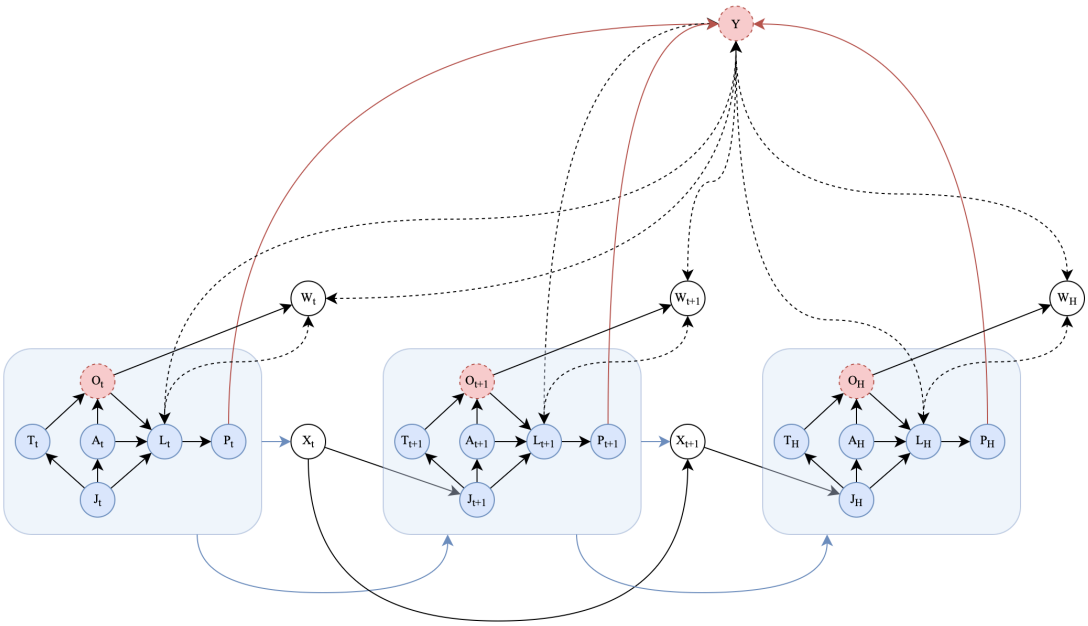


Figure 5: AntMaze

C Implementation Details

This section summarizes the implementation choices common to all experiments in CILBench, including environment construction, expert training, causal adjustment, model architectures, and optimization settings. All experiments were conducted on a machine equipped with an NVIDIA H100, using PyTorch 2.4.0.

Environment Wrapper and SCM Integration. Each OGBench environment is wrapped in a Pearl Causal Hierarchy (PCH) interface that exposes `see()` for observational rollouts, `do()` for interventional evaluation, and accessors for internal endogenous variables. Latent confounders are injected via a structural equation for an exogenous wind field whose dynamics are governed by a piecewise-constant stochastic process with refresh interval 5 steps. Partially observed variables (e.g., heading measurements) are computed as noisy functions of both observable state components and latent confounders. All confounders and auxiliary measurements are integrated into the causal graph extracted from the environment at initialization.

Expert Training. Experts are obtained through a two-stage pipeline. First, we train an offline behavioral cloning policy on the OGBench dataset using a fully observed state representation. The BC policy uses a residual MLP with hidden dimension 256, activation `SiLU`, and 4 residual blocks. Second, the BC policy is fine-tuned under the confounded environment using a TD3-style actor–critic algorithm with target smoothing coefficient 5×10^{-3}, policy delay 2, discount factor 0.99, and batch size 256. Fine-tuning is run for $300,000$ environment steps or until convergence. The resulting expert is then used to collect demonstrations via `env.see()` using deterministic execution.

Sequential Adjustment and Windowed Encoding. Adjustment sets are computed using the windowed sequential π-backdoor procedure (Algorithm 2) with lookback $k \in \{1, 2, 3, 5\}$. The proxy graph is constructed from a short-horizon environment of length $k + 1$, and the base adjustment sets are shifted along the full horizon and trimmed to retain only variables within the last k steps. For each observed variable V with dimension d_V, we construct window slots of shape (k, d_V) and represent each time-t encoding vector z_t as the concatenation of all selected slots, yielding input dimension 42 for the actor, critic, and discriminator. Continuous variables are normalized using running statistics computed from expert trajectories; categorical variables are one-hot encoded.

Network Architectures. All imitation policies (BC or GAIL, causal or naive) use the same family of networks to ensure architectural parity.

- **Actor**: a residual MLP based on `ContinuousActor`, with hidden size 256, depth 3 residual blocks, dropout 0.05, and `SiLU` activations. Actions are parameterized via a Gaussian with state-independent log-variance and squashed through `tanh` to match environment bounds.
- **Critic**: identical residual architecture mapping z_t to a scalar value estimate.
- **Discriminator**: a residual MLP of hidden size 256 applied to concatenated (z_t, a_t) pairs; trained with BCE or WGAN-style losses depending on the experiment.

Behavioral Cloning. Causal BC is trained with Huber loss and Adam optimizer (learning rate 3×10^{-4}, batch size 2048, early stopping patience 15). Naive BC conditions on all observable variables; causal BC uses only windowed adjustment features. Training runs for 1000 epochs with validation split 80/20.

GAIL Training. Causal and naive GAIL share identical optimization hyperparameters:

- **Rollouts**: each GAIL round collects 10 episodes up to horizon $H = 1000$ using stochastic actor sampling.

- **GAE/PPO**: advantages computed with $(\gamma, \lambda) = (0.99, 0.95)$; PPO uses clip ratio 0.2, minibatch size 1024, and 10 epochs per update.

- **Discriminator**: updated 3 times per round with minibatch size 1024, gradient penalty weight 10.0, and instance noise standard deviation 0.0.

All networks are optimized with Adam (learning rates: actor 1×10^{-4}, critic 3×10^{-4}, discriminator 3×10^{-4}).

Evaluation. Policies are evaluated in interventional mode using `env.do()`, with deterministic actions for BC and with mean-action execution for GAIL. Each reported metric averages over 1000 episodes with fixed seeds. Additional visualizations (state trajectories, heatmaps, return distributions) are provided in Appendix D.

All code, including environment wrappers, adjustment-set utilities, and training pipelines for BC and GAIL, is packaged with CILBench to support future extensions and reproducibility.

D Additional Visualizations

This appendix provides supplementary visualizations that complement the quantitative results in Section 4. We include learning dynamics, trajectory summaries, return distributions, and success metrics for both AntMaze-Medium and AntMaze-Large. All evaluations are performed in interventional mode using the PCH wrapper.

D.1 Return Distributions

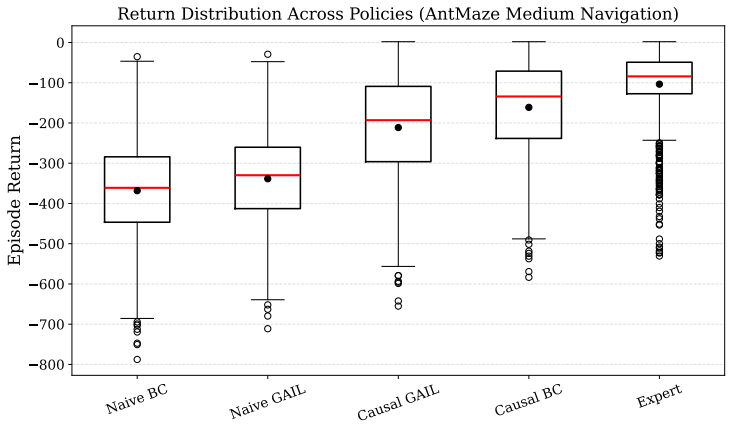


Figure 6: Distribution of raw episode returns across algorithms on AntMaze-Medium. Causal methods achieve significantly higher and more stable returns than naive baselines.

D.2 Success Lengths

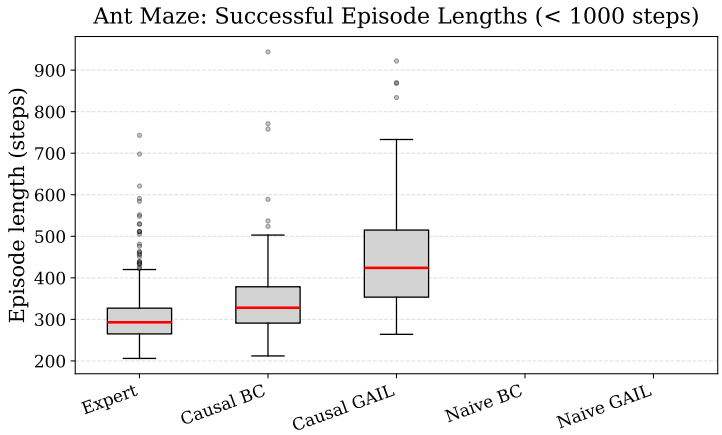


Figure 7: Distribution of episode lengths among successful rollouts in AntMaze-Medium. The expert achieves the shortest paths; causal BC and GAIL follow with slightly longer trajectories; naive methods have no successful episodes.

D.3 Success Rates

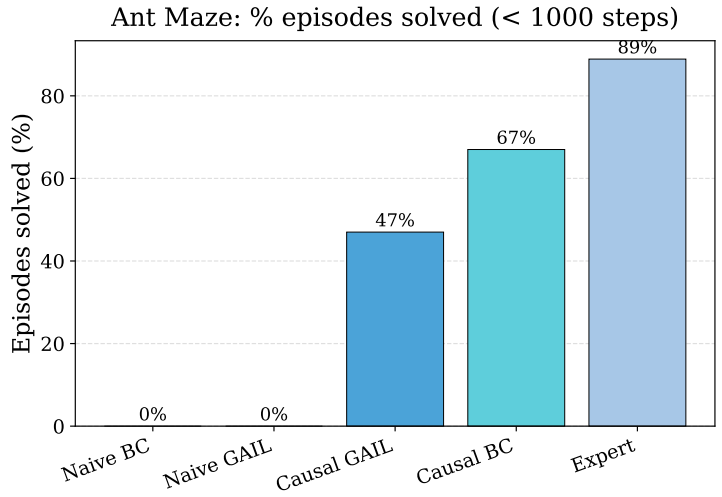


Figure 8: Success rate across 1000 evaluation episodes for AntMaze-Medium. Causal BC and Causal GAIL significantly outperform naive baselines, which never succeed.

D.4 Discriminator Diagnostics

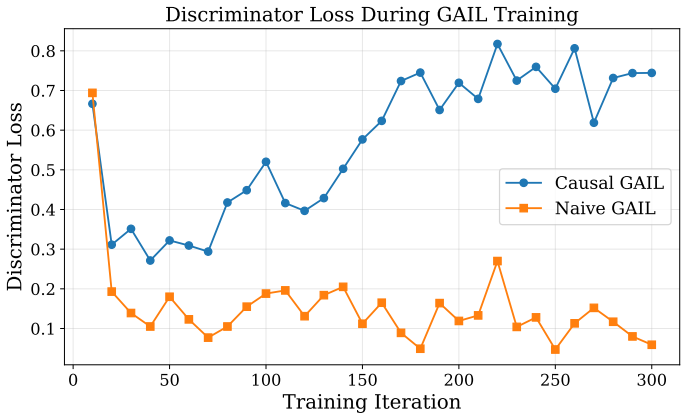


Figure 9: Discriminator loss during GAIL training on AntMaze-Medium. Naive GAIL exhibits a lower discriminator loss than Causal GAIL, suggesting overfitting to spurious features created by latent confounding. In contrast, the causal representation forces the discriminator to attend only to deconfounded state information, preventing collapse.