Sequential Causal Imitation Learning with Unobserved Confounders

Anonymous Author(s)
Affiliation
Address
email

Abstract

"Monkey see monkey do" is an age-old adage, referring to naïve imitation without a deep understanding of a system's underlying mechanics. Indeed, if a demonstrator has access to information unavailable to the imitator (monkey), such as a different set of sensors, then no matter how perfectly the imitator models its perceived environment (SEE), attempting to directly reproduce the demonstrator's behavior (DO) can lead to poor outcomes. Imitation learning in the presence of a mismatch between demonstrator and imitator has been studied in the literature under the rubric of causal imitation learning (Zhang et al., 2020), but existing solutions are limited to single-stage decision-making. This paper investigates the problem of causal imitation learning in sequential settings, where the imitator must make multiple decisions per episode. We develop a graphical criterion that is both necessary and sufficient for determining the feasibility of causal imitation, providing conditions when an imitator can match a demonstrator's performance despite differing capabilities. Finally, we provide an efficient algorithm for determining imitability, and corroborate our theory with simulations.

1 Introduction

Without access to observational data, an agent must learn how to operate at a suitable level of performance through trial and error (Sutton et al., 1998; Mnih et al., 2013). This from-scratch approach is often impractical in environments with the potential of extreme negative - and final - outcomes (driving off a cliff). While both Nature and machine learning researchers have approached the problem from a wide variety of perspectives, a particularly potent method which has been used with great success in many learning machines, including humans, is exploiting observations of other agents in the environment (Rizzolatti & Craighero, 2004; Hussein et al., 2017).

Learning to act by observing other agents offers a data multiplier, allowing agents to take into account others' experiences. Even when the precise loss function is unknown (what exactly goes into being a good driver?), the agent can attempt to learn from "experts", namely agents which are known to gain an acceptable reward at the target task. This approach has been studied under the umbrella of *imitation learning* (Argall et al., 2009; Billard et al., 2008; Hussein et al., 2017; Osa et al., 2018). Several methods have been proposed, including *inverse reinforcement learning* (Ng et al., 2000; Abbeel & Ng, 2004; Syed & Schapire, 2008; Ziebart et al., 2008) and *behavior cloning* (WIDROW, 1964; Pomerleau, 1989; Muller et al., 2006; Mülling et al., 2013; Mahler & Goldberg, 2017). The former attempts to reconstruct the loss/reward function that the experts minimize and then use it for optimization; the latter directly copies the expert's actions (behavior cloning).

Despite the power entailed by this approach, it relies on a somewhat stringent condition: the expert and imitator's sensory capabilities need to be perfectly matched. As an example, self-driving cars rely solely on cameras or lidar, completely ignoring the auditory dimension - and yet most human

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

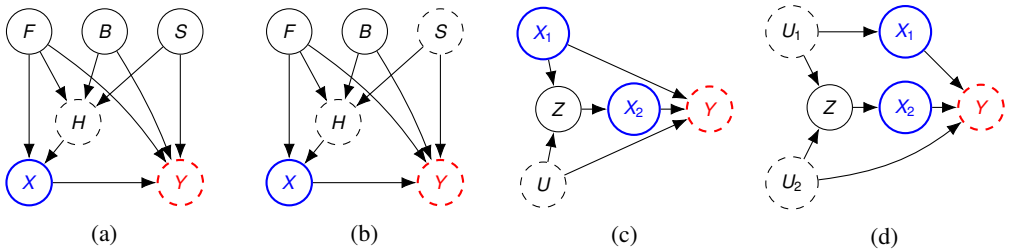


Figure 1: (a, b) represents a simplified view of a driver X and surrounding cars F, B, S. (c) is imitable with policies $\pi_1(X_1) = P(X_1)$ and $\pi_2(X_2|Z) = P(X_2|Z)$, but in (d) X_1, X_2 is not imitable, despite there being a valid sequential backdoor.

demonstrators are able to exploit this data, especially in dangerous situations (car horns, screeching tires). Perhaps without a microphone, the self-driving car would incorrectly attribute certain behaviors to visual stimuli, leading to a poor policy? For concreteness, consider the scenario shown in fig. 1a, where the human driver (X, i.e., the demonstrator, in blue) is looking forward (F), and can hear car horns (H) from cars behind (B), and to the side (S). The driver's performance is represented by a variable Y (red), which is unobserved (dashed node). Since our dataset only contains visual data, car horns H remain unobserved to the learning agent (i.e., the imitator). Despite not being able to hear car horns, the learner from Fig. 1a had a full view of the car's surroundings, including cars behind and to the side, which turns out to be sufficient to perform imitation in this example. To witness, consider an instance where F, B, S are drawn uniformly over $\{0, 1\}$. The reward Y is decided by $\neg X \oplus F \oplus B \oplus S$; $\oplus$ represents the *exclusive-or* operator. The human driver decides the action $X \leftarrow H$ where values of horn H is given by $F \oplus B \oplus S$. Preliminary analysis reveals that the learner could perfectly mimic the demonstrator's decision-making process using an imitating policy $X \leftarrow F \oplus B \oplus S$. On the other hand, if the driving system does not have side cameras, the side view S becomes latent; see Fig. 1b. The learner's reward $\mathbf{E}[Y|\text{do}(\pi)]$ is equal to 0.5 for any policy $\pi(x|f, b)$, which is far from the optimal demonstrator's performance, $\mathbf{E}[Y] = 1$.

Based on these examples, there arises the question of determining precise conditions under which an agent can account for the lack of knowledge or observations available to the expert, and how this knowledge should be combined to generate an optimal imitating policy, giving identical performance as the expert on measure Y. These questions have been recently investigated in the context of *causal imitation learning* (Zhang et al., 2020), where a complete graphical condition and algorithm were developed for determining imitability in the single-stage decision-making setting with partially observable models (i.e., in non-Markovian settings). Other structural assumptions, such as linearity (Etesami & Geiger, 2020), were also explored in the literature, but were still limited to the single-stage setting. Despite all this progress, there are still significant challenges in undertaking causal imitation for the general case. Existing methods only allow for proper causal imitation when a single action X is considered per episode (e.g., Fig. 1a), and it is unclear how to systematically determine how to imitate, or even whether imitation is possible when a learner must make several actions in sequence (e.g., Figs. 1c and 1d).

The goal of this paper is to fill this gap in understanding. More specifically, our contributions are as follows. (1) We provide a graphical criterion for determining whether imitability is feasible in sequential settings based on a causal graph encoding the domain's causal structure. (2) We propose an efficient algorithm to determine imitability and to find the policy for each action that leads to proper imitation. (3) We prove that the proposed criterion is complete (i.e. both necessary and sufficient). Finally, we experimentally verify that our approach compares favorably with existing methods in contexts where a demonstrator has access to latent variables. Due to space constraints, proofs are provided in the appendix.

1.1 Preliminaries

We start by introducing the notation and definitions used throughout the paper. In particular, we use capital letters for random variables (Z), and small letters for their values (z). Bolded letters represent sets of random variables and their samples ($\boldsymbol{Z} = \{Z_1, ..., Z_n\}$, $\boldsymbol{z} = \{z_1 \sim Z_1, ..., z_n \sim Z_n\}$).

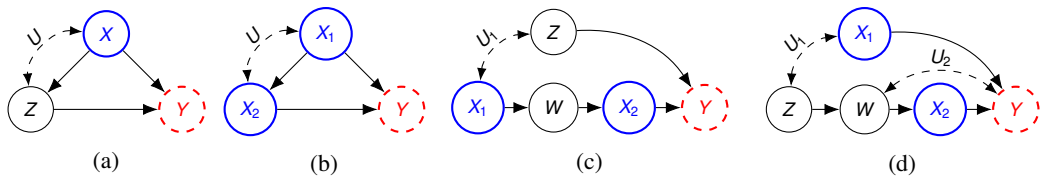


Figure 2: Despite there being no latent path between Y and any X, the query in (a) is not imitable, but the query in (b) is imitable. While (c) is imitable if Z comes before X_2 in temporal order, the query in (d) is imitable only if Z comes before X_1.

$|\boldsymbol{Z}|$ represents a set's cardinality. To simplify notation, we consistently use the shorthand $P(z_i)$ to represent probabilities $P(Z_i = z_i)$.

The basic semantic framework of our analysis rests on *structural causal models* (SCMs) (Pearl, 2000, Ch. 7). An SCM M is a tuple $\langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F}, P(\boldsymbol{u}) \rangle$ with $\boldsymbol{V}$ the set of endogenous, and $\boldsymbol{U}$ exogenous variables. $\boldsymbol{F}$ is a set of structural functions s.t. for $f_V \in \boldsymbol{F}$, $V \leftarrow f_V(pa(V), U_V), pa(V) \subseteq \boldsymbol{V}, U_V \subseteq \boldsymbol{U}$. Values of $\boldsymbol{U}$ are drawn from an exogenous distribution $P(\boldsymbol{u})$, inducing distribution $P(\boldsymbol{v})$ over the endogenous $\boldsymbol{V}$. Since the learner can observe only a subset of endogenous variables, we split $\boldsymbol{V}$ into $\boldsymbol{O} \subseteq \boldsymbol{V}$ (observed) and $\boldsymbol{L} = \boldsymbol{V} \setminus \boldsymbol{O}$ (latent) sets of variables. The marginal $P(\boldsymbol{o})$ is thus referred to as the *observational distribution*.

Each SCM M is associated with a causal diagram $\mathcal{G}$ where (e.g., see Fig. 2d) solid nodes represent observed variables $\boldsymbol{O}$, dashed nodes represent latent variables $\boldsymbol{L}$, and arrows represent the arguments $pa(V)$ of each functional relationship f_V. Exogenous variables $\boldsymbol{U}$ are not explicitly shown; a bi-directed arrow between nodes V_i and V_j indicates the presence of an unobserved confounder (UC) affecting both V_i and V_j. We will use standard conventions to represent graphical relationships such as parents, children, descendants, and ancestors. For example, the set of parent nodes of $\boldsymbol{X}$ in $\mathcal{G}$ is denoted by $pa(\boldsymbol{X})_{\mathcal{G}} = \cup_{X \in \boldsymbol{X}} pa(X)_{\mathcal{G}}$. ch, de and an are similarly defined. Capitalized versions Pa, Ch, De, An include the argument as well, e.g. $De(\boldsymbol{X})_{\mathcal{G}} = de(\boldsymbol{X})_{\mathcal{G}} \cup \boldsymbol{X}$. An observed variable $V_i \in \boldsymbol{O}$ is an *effective parent* of $V_j \in \boldsymbol{V}$ if there is a directed path from V_i to V_j in $\mathcal{G}$ such that every internal node on the path is in $\boldsymbol{L}$. We define $pa^+(\boldsymbol{S})$ as the set of effective parents of variables in $\boldsymbol{S}$, excluding $\boldsymbol{S}$ itself, and $Pa^+(\boldsymbol{S})$ as $\boldsymbol{S} \cup pa^+(\boldsymbol{S})$. Other relations, like $ch^+(\boldsymbol{S})$ are defined similarly.

A path from a node X to a node Y in $\mathcal{G}$ is said to be "active" conditioned on a (possibly empty) set $\boldsymbol{W}$ if it contains a collider ($\rightarrow A \leftarrow$) only if $A \in An(\boldsymbol{W})$, and does not otherwise contain vertices from $\boldsymbol{W}$ (d-separation, Koller & Friedman (2009)). $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent conditioned on $\boldsymbol{W}$ $(\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{W})_{\mathcal{G}}$ if there are no active paths between any $X \in \boldsymbol{X}$ and $Y \in \boldsymbol{Y}$. For a subset $\boldsymbol{X} \subseteq \boldsymbol{V}$, the subgraph obtained from $\mathcal{G}$ with edges outgoing from $\boldsymbol{X}$ / incoming into $\boldsymbol{X}$ removed is written $\mathcal{G}_{\underline{\boldsymbol{X}}}/\mathcal{G}_{\overline{\boldsymbol{X}}}$ respectively. Finally, we utilize a special type of clustering of observed nodes in a causal diagram, called *confounded components* (Tian & Pearl, 2002; Tian, 2002).

Definition 1.1. *For a causal diagram $\mathcal{G}$, let $\boldsymbol{N}$ be a set of unobserved variables in $\boldsymbol{L} \cup \boldsymbol{U}$. A set $\boldsymbol{C} \subseteq Ch(\boldsymbol{N}) \cap \boldsymbol{O}$ is a c-component if for any pair $U_i, U_j \in \boldsymbol{N}$, there exists a path between U_i and U_j in $\mathcal{G}$ such that every observed node $V_k \in \boldsymbol{O}$ on the path is a collider (i.e., $\rightarrow V_k \leftarrow$).*

In particular, we focus on *maximal* c-components $\boldsymbol{C}$, where there doesn't exist c-component $\boldsymbol{C}'$ s.t. $\boldsymbol{C} \subset \boldsymbol{C}'$. The collection of maximal c-components forms a partition $\boldsymbol{C}_1, \ldots, \boldsymbol{C}_m$ over observed variables $\boldsymbol{O}$. For any set $S \subseteq \boldsymbol{O}$, let $\boldsymbol{C}(S)$ be the union of c-components $\boldsymbol{C}_i$ that contain variables in S. For instance, for variable Z in Fig. 1d, the c-component $\boldsymbol{C}(\{Z\}) = \{Z, X_1\}$.

2 Causal Sequential Imitation Learning

We are interested in learning a policy over a series of actions $\boldsymbol{X} \subseteq \boldsymbol{O}$ so that an imitator gets average reward $Y \in \boldsymbol{V}$ identical to that of an expert demonstrator. More specifically, let variables in $\boldsymbol{X}$ be ordered by $X_1, \ldots, X_n$, $n = |\boldsymbol{X}|$. Actions are taken sequentially by the imitator, where only information available at the time of the action can be used to inform a policy for $X_i \in \boldsymbol{X}$. To encode the ordering of observations and actions in time, we fix a topological ordering on the variables of $\mathcal{G}$, which we call the "temporal ordering". We define functions $\text{before}(X_i)$ and $\text{after}(X_i)$ to represent nodes that come before/after an action $X_i \in \boldsymbol{X}$ following the ordering, excluding X_i itself. A policy π on actions $\boldsymbol{X}$ is a sequence of decision rules $\{\pi_1, \ldots, \pi_n\}$ where each $\pi_i(x_i | \boldsymbol{z}_i)$ is a function

mapping from domains of covariates $\boldsymbol{Z}_i \subseteq \text{before}(X_i)$ to the domain of action X_i. The imitator following a policy π replacing the demonstrator in an environment is encoded by replacing the expert's original policy in the SCM M with π, which gives the results of the imitator's actions as $P(\boldsymbol{v}|\text{do}(\pi))$. Our goal is to learn an imitating policy π such that the induced distribution $P(y|\text{do}(\pi))$ perfectly matches the original expert's performance $P(y)$. Formally

Definition 2.1. *(Zhang et al., 2020) Given a causal diagram $\mathcal{G}$, $\boldsymbol{Y} \subseteq \boldsymbol{V}$ is said to be imitable with respect to $\boldsymbol{X} \subseteq \boldsymbol{O}$ in $\mathcal{G}$ if there exists $\pi \in \Pi$ uniquely computable from the observational distribution $P(\boldsymbol{o})$ such that for all possible SCMs M compatible with $\mathcal{G}$, $P(\boldsymbol{Y})_M = P(\boldsymbol{Y}|do(\pi))_M$.*

For single stage decision-making problems ($\boldsymbol{X} = \{X\}$), Zhang et al. (2020) demonstrated imitability for reward Y if and only if there exists a set of covariates $\boldsymbol{Z} \in \text{before}(X)$ such that $(Y \perp\!\!\!\perp X|\boldsymbol{Z})_{\mathcal{G}_{\underline{X}}}$, called the *backdoor admissible set* (Pearl, 2000, Def. 3.3.1) ($\boldsymbol{Z} = \{F, B, S\}$ in Fig. 1a).

Since the backdoor criterion is complete for the single-stage problem, one may be tempted to surmise that a version of the criterion generalized to multiple interventions might likewise solve the imitability problem in the general case ($|\boldsymbol{X}| > 1$). Pearl & Robins (1995) generalized the backdoor criterion to the sequential setting as follows:

Definition 2.2. *(Pearl & Robins, 1995) Given a causal diagram $\mathcal{G}$, a set of action variables $\boldsymbol{X}$, and target node Y, sets $\boldsymbol{Z}_1 \subseteq \text{before}(X_1), \ldots, \boldsymbol{Z}_n \subseteq \text{before}(X_n)$ satisfy the sequential backdoor for $(\mathcal{G}, \boldsymbol{X}, Y)$ if for each $X_i \in \boldsymbol{X}$ such that $(Y \perp\!\!\!\perp X_i|\boldsymbol{X}_{1:i-1}, \boldsymbol{Z}_{1:i})_{\mathcal{G}_{\underline{X}_i \overline{\boldsymbol{X}}_{i+1:n}}}$.*

where $X_{i:j} = \{X_i, X_{i+1}, \ldots, X_j\}$. While the sequential backdoor is an extension of the backdoor to multi-stage decisions, its existence does not always guarantee the imitability of latent reward Y. In Fig. 1d, $\boldsymbol{Z}_1 = \{\}, \boldsymbol{Z}_2 = \{Z\}$ is a sequential backdoor set for $(\mathcal{G}, \{X_1, X_2\}, Y)$, but there are distributions for which no agent can imitate the demonstrator's performance (Y) without knowledge of either the latent U_1 or U_2. To witness, suppose that the adversary sets up an SCM with binary variables as follows: $U_1, U_2 \sim Bern(0.5)$, with $X_1 := U_1$, $Z := U_1 \oplus U_2$, $X_2 := Z$ and $Y = \neg(X_1 \oplus X_2 \oplus U_2)$, with $\oplus$ as a binary XOR. The fact that $U \oplus U = 0$ is exploited to generate a chain where each latent variable appears exactly twice in Y, making $Y = \neg(U_1 \oplus (U_1 \oplus U_2) \oplus U_2) = 1$. On the other hand, when imitating, X_1 can no longer base its value on U_1, making the imitated $\hat{Y} = \neg(\hat{X}_1 \oplus \hat{X}_2 \oplus U_2)$. Since the imitator only knows Z, it can do no better than $\mathbf{E}[\hat{Y}] = 0.5$ (For a more detailed explanation, we refer readers to Prop. C.1)!

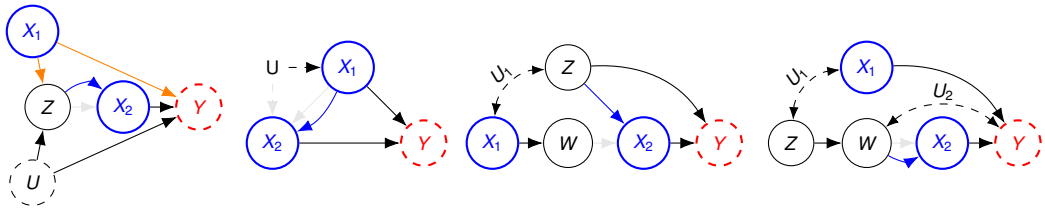
2.1 Sequential Backdoor for Causal Imitation

We now introduce the main result of this paper: a generalized backdoor criterion that allows one to learn imitating policies in the sequential setting. For a sequence of covariates $\boldsymbol{Z}_1 \subseteq \text{before}(X_1), \ldots, \boldsymbol{Z}_n \subseteq \text{before}(X_n)$, let $\mathcal{G}'_i$, $i = 1, \ldots, n$, be the manipulated graph obtained from a causal diagram $\mathcal{G}$ by first (1) removing all arrows coming into nodes in $X_{i+1:n}$; and (2) adding arrows $\boldsymbol{Z}_{i+1} \to X_{i+1}, \ldots, \boldsymbol{Z}_n \to X_n$. We can then define a sequential backdoor criterion for causal imitation as follows:

Definition 2.3. *Given a causal diagram $\mathcal{G}$, a set of action variables $\boldsymbol{X}$, and target node Y, sets $\boldsymbol{Z}_1 \subseteq \text{before}(X_1), \ldots, \boldsymbol{Z}_n \subseteq \text{before}(X_n)$ satisfy the **"sequential π-backdoor"** for $(\mathcal{G}, \boldsymbol{X}, Y)$ if at each $X_i \in \boldsymbol{X}$, either (1) $(X_i \perp\!\!\!\perp Y|\boldsymbol{Z}_i)$ in $(\mathcal{G}'_i)_{\underline{X_i}}$, or (2) $X_i \notin An(Y)$ in $\mathcal{G}'_i$.*

The first condition of Def. 2.3 is similar to the standard backdoor criterion where $\boldsymbol{Z}_i$ is a set of variables that effectively encodes all of the information relevant to imitating X_i with respect to Y. In other words, if the joint $P(\boldsymbol{Z}_i \cup \{X_i\})$ matches when both expert and imitator are acting, then Y cannot tell the difference. The critical modification of the original π-backdoor for the sequential setting comes from the causal graph in which this check happens. $\mathcal{G}'_i$ can be seen as $\mathcal{G}$ with all future actions of the imitator already encoded in the graph. That is, when performing a check for X_i, it is done with all actions after i being performed by the imitator rather than expert, with the associated parents of each future $X_{j>i}$ replaced with their corresponding imitator's conditioning set. Several examples of $\mathcal{G}'_i$ are shown in Fig. 3.

The second condition allows for the case where an action at X_i has no effect on the value of Y once future actions are taken. Since $\mathcal{G}'_i$ has modified parents for future $\boldsymbol{X}_{j>i}$, the value of X_i might no longer be relevant at all to Y, i.e. Y would get the same input distribution no matter what policy is chosen for X_i. This allows X_i to fail condition (1), meaning that it is not imitable by itself, but still be part of an imitable set $\boldsymbol{X}$, because future actions can "correct" for the errors made at X_i.



(a) (b) (c) (d)

Figure 3: Examples of $\mathcal{G}'_1$. In Fig. 1c, we can have $\boldsymbol{Z}_1 = \emptyset, \boldsymbol{Z}_2 = \{Z\}$, so X_2 has its parents cut, and a new arrow added from Z to X_2 (blue). The independence check $(X_1 \perp\!\!\!\perp Y|\emptyset)$ is done in graph (a) with edges outgoing from X_1 removed (orange). In Fig. 2b, using $\boldsymbol{Z}_1 = \emptyset, \boldsymbol{Z}_2 = \{X_1\}$, we first replace the parents of X_2 with just X_1 (b), and then remove both resulting outgoing edges from X_1 to check if $(X_1 \perp\!\!\!\perp Y)$. On the other hand, in Fig. 2c, if $\boldsymbol{Z}_2 = \{Z\}$, we get (c), which means $X_i \notin An(Y)$, passing condition 2 of Def. 2.3. Finally, in Fig. 2d, with $\boldsymbol{Z}_2 = \{W\}$, X_1 must condition on either Z or W to be independent of Y in (d) once the edge $X_1 \to Y$ is removed.

The distinction between condition 1 and condition 2 is shown in Fig. 3c: in the original graph $\mathcal{G}$ described in Fig. 2c, if Z comes after X_1, then there is no valid conditioning set that can d-separate X_1 from Y. However, if the imitating policy for X_2 uses Z instead of W or X_1 (i.e. $\pi_{X_2} = P(X_2|Z)$), X_1 will no longer be an ancestor of Y in $\mathcal{G}'_1$. In effect, the action made at X_2 shields Y from inevitable mistakes made at X_1 due to not having access to confounder U_1 when taking the action.

Indeed, the sequential π-backdoor criterion can be seen as a recursively applying the single-action π-backdoor. Starting from the last action X_k in temporal order, one can directly show that Y is imitable using a backdoor set $\boldsymbol{Z}_k$ (or X_k doesn't affect Y by any causal path). Replacing X_k in the SCM with this new imitating policy, the resulting SCM with graph G'_{k-1} has an identical distribution over Y as $\mathcal{G}$. The procedure can then be repeated for X_{k-1} using G'_{k-1} as the starting graph, and continued recursively, showing imitability for the full set:

Theorem 2.1. *Given a causal diagram $\mathcal{G}$, a set of action variables $\boldsymbol{X}$, and target node Y, if there exist sets $\boldsymbol{Z}_1, \boldsymbol{Z}_2, ..., \boldsymbol{Z}_k$ that satisfy the sequential π-backdoor criterion with respect to $(\mathcal{G}, \boldsymbol{X}, Y)$, then Y is imitable with respect to $\boldsymbol{X}$ in $\mathcal{G}$ with policy $\pi_{X_i}(\boldsymbol{Z_i}) = P(X_i|\boldsymbol{Z}_i)$ for each $X_i \in \boldsymbol{X}$.*

Thm. 2.1 establishes the sufficiency of the sequential π-backdoor for imitation learning. For instance, consider again the diagram in Fig. 2c. It is verifiable that the covariate set $\boldsymbol{Z}_1 = \{\}, \boldsymbol{Z}_2 = \{Z\}$ is sequential π-backdoor admissible. Thm. 2.1 implies that the imitating policy is given by $\pi_1(x_1) = P(x_1)$ and $\pi_2(x_2|z) = P(x_2|z)$. Once π-backdoor admissible sets are obtained, the imitating policy can be learned from the observational data through standard density estimation methods for stochastic policies, and supervised learning methods for deterministic policies.

3 Finding Sequential π-Backdoor Admissible Sets

The recursive nature of Def. 2.3 suggests a natural algorithm for finding a sequence of covariates $\boldsymbol{Z}_{1:n}$ that satisfy the sequential π-backdoor condition. Let a collection $\boldsymbol{Z}_{i:n}$, $i = 1, \ldots, n$, be sequential $\pi_{>i}$ backdoor admissible if it satisfies conditions in Def. 2.3 for actions in $X_i, X_{i+1}, \ldots, X_n$. Given a sequential $\pi_{>i+1}$ backdoor admissible set $\boldsymbol{Z}_{i+1:n}$, $i = 1, \ldots, n$, we could find all sequential $\pi_{>i}$ backdoor admissible sets by listing all covariate sets $\boldsymbol{Z}_i$ that are backdoor admissible w.r.t. the single action X_i at the i-th stage. Several efficient methods for finding such back-door conditioning sets have been developed in the literature (Tian & Paz, 1998; van der Zander & Liśkiewicz, 2020). Recursively applying this operation for all actions following a reverse temporal ordering over $\boldsymbol{X}$ eventually leads to a sequential π-backdoor admissible set $\boldsymbol{Z}_{1:n}$.

However, for every single action $X_i \in \boldsymbol{X}$, there could be exponentially many backdoor admissible sets $\boldsymbol{Z}_i$. Since the sequential π-backdoor admissibility of a covariate set $\boldsymbol{Z}_i$, $i = 1, \ldots, n-1$, depends on all the other covariate sets $\boldsymbol{Z}_{i+1}, \ldots, \boldsymbol{Z}_n$ coming after it, one may have to check all the backdoor admissible sets $\boldsymbol{Z}_i$ for every action $X_i \in \boldsymbol{X}$, which is not feasible in practical settings. To address these issues, this section will see the development of Alg. 1, which efficiently finds a valid sequential π-backdoor admissible set $\boldsymbol{Z}_{1:n}$ with regard to actions $\boldsymbol{X}$ in a causal diagram $\mathcal{G}$, if such a set exists.

Algorithm 1 Find largest valid $\boldsymbol{O}^X$ in ancestral graph of Y given $\mathcal{G}$, $\boldsymbol{X}$ and target Y

function HASVALIDCONDITIONING($\mathcal{G}$,$\boldsymbol{O}^X$,O_i,X_i)
  $\boldsymbol{C} \leftarrow$ the c-component of O_i
  $\mathcal{G}_{\boldsymbol{C}} \leftarrow$ the subgraph of $\mathcal{G}$ containing only $Pa^+(\boldsymbol{C})$ and intermediate latent variables
  return $(V_i \perp\!\!\!\perp \boldsymbol{C} \setminus (\boldsymbol{O}^X \cup \{O_i\}) | (\boldsymbol{C} \setminus (\boldsymbol{O}^X \cup \{V_i\})) \cap \text{before}(X_i))$ in $\mathcal{G}_{\boldsymbol{C}}$
function FINDOX($\mathcal{G}$,$\boldsymbol{X}$,Y)
  $\mathscr{O}^X \leftarrow$ empty map from elements of $\boldsymbol{O}$ to elements of $\boldsymbol{X}$
  do
    for $O_i \in \boldsymbol{O}$ of $\mathcal{G}^Y$ (ancestral graph of Y) in reverse temporal order **do**
      if $|ch^+(O_i)| > 0$ **and** $ch^+(O_i) \subseteq keys(\mathscr{O}^X)$ **then**
        $X_i \leftarrow$ earliest element of $\mathscr{O}^X[ch^+(O_i)]$ in temporal order
        if HASVALIDCONDITIONING($\mathcal{G}$,$keys(\mathscr{O}^X)$,O_i,X_i) **then**
          $\mathscr{O}^X[O_i] \leftarrow X_i$
      else if $V_i \in \boldsymbol{X}$ **and** HASVALIDCONDITIONING($\mathcal{G}$,$keys(\mathscr{O}^X)$,O_i,O_i) **then**
        $\mathscr{O}^X[O_i] \leftarrow O_i$
  while $|\mathscr{O}^X|$ changed in most recent pass
  return $keys(\mathscr{O}^X)$

To create the relevant conditioning sets, we will use a Markov Boundary (minimal Markov Blanket, Pearl (1988)) for a set of nodes $\boldsymbol{O}^X \subseteq \boldsymbol{O}$, which is defined as the minimal set $\boldsymbol{Z} \subset \boldsymbol{O} \setminus \boldsymbol{O}^X$ such that $(\boldsymbol{O}^X \perp\!\!\!\perp \boldsymbol{O} \setminus \boldsymbol{O}^X | \boldsymbol{Z})$. This definition can be applied to graphs with latent variables, where it can be constructed in terms of c-components:

Lemma 3.1. *Given $\boldsymbol{O}^X \subseteq \boldsymbol{O}$, the Markov Boundary of $\boldsymbol{O}^X$ in $\boldsymbol{G}$ is $Pa^+(\boldsymbol{C}(Ch^+(\boldsymbol{O}^X))) \setminus \boldsymbol{O}^X$*

To see the utility of the Markov Boundary, consider that if there is a set $\boldsymbol{Z}$ that satisfies the single π-backdoor for X_i, then taking $\mathcal{G}^Y$ as the ancestral graph of Y, the Markov Boundary $\boldsymbol{Z}'$ of X_i in $\mathcal{G}^Y_{\underline{X_i}}$ is also a valid π-backdoor, because $(Y \perp\!\!\!\perp X_i | \boldsymbol{Z}')$ in $\mathcal{G}^Y_{\underline{X_i}}$ by definition, and $\boldsymbol{Z}' \subseteq \text{before}(X_i)$ because with outgoing edges from X_i removed, the boundary simplifies to $Pa^+(\boldsymbol{C}(X_i)) \setminus \{X_i\}$, and in the ancestral graph of Y, each element of $\boldsymbol{C}(X_i)$ is an ancestor of Y, and so has an element of $\boldsymbol{Z} \subseteq \text{before}(X_i)$ blocking each such path - and therefore $Pa^+(\boldsymbol{C}(X_i)) \subseteq \text{before}(X_i)$ too. In other words, the Markov Boundary is a good candidate method for generating conditioning sets for imitation that will satisfy the requirements of the sequential π-backdoor.

Nevertheless, a naïve algorithm that uses the Markov Boundary of $X_i \in \boldsymbol{X}$ in $(\mathcal{G}'_i)^Y_{\underline{X_i}}$ as the corresponding $\boldsymbol{Z}_i$, and returns a failure whenever $\boldsymbol{Z}_i \notin \text{before}(X_i)$ for the sequential π-backdoor still has all of the weaknesses described above. It cannot create a valid sequential π-backdoor for Fig. 2c, since X_2 would have $\boldsymbol{Z}_2 = \{W\}$, but no conditioning set exists for X_1 that d-separates it from Y in $\mathcal{G}'_1$.

To mitigate this issue, we notice that an X_i does not require a valid conditioning set if it is not an ancestor of Y in $\mathcal{G}'_i$ (i.e. X_i does not need to satisfy (1) of Def. 2.3 if it can satisfy (2)). Furthermore, even if X_i is an ancestor of Y, and therefore must satisfy condition (1), any elements of its c-component that are not ancestors of Y in $\mathcal{G}'_i$ won't be part of $(\mathcal{G}'_i)^Y$, effectively splitting the c-component in two, making it more likely that the variables in the boundary set $\boldsymbol{Z}_i$ in the component containing X_i be in $\text{before}(X_i)$. It is therefore beneficial for an action X_i to have a conditioning set that uses the earliest variables possible in temporal order, so that actions $X_{j<i}$ have maximized chance of satisfying (2), and have the smallest possible c-components in $\mathcal{G}'_i$.

FINDOX in Alg. 1 finds a set $\boldsymbol{O}^X \subseteq \boldsymbol{O}$ of ancestors of $\boldsymbol{X}$ (and including $\boldsymbol{X}$) in $\mathcal{G}^Y$ that do not need to be conditioned by *any* $\boldsymbol{X}$. Elements of this set (possibly excluding $\boldsymbol{X}$) will not be ancestors of Y once the actions in their descendants are taken. That is, an element $O_i \in \boldsymbol{O}^X$ where $ch^+(O_i) \in \boldsymbol{O}^X$ is not present in $\mathcal{G}'_i$ for all actions that come before it in temporal order, and can therefore effectively be ignored. Before showing examples, we verify that the set $\boldsymbol{O}^X$ returned by FINDOX can be used to construct a sequential π-backdoor:

Definition 3.1. *The set $\boldsymbol{X}^B \subseteq \boldsymbol{X}$ called the "boundary actions" for $\boldsymbol{O}^X :=$ FINDOX$(\mathcal{G}, \boldsymbol{X}, Y)$ are all elements X_i of $\boldsymbol{O}^X$ where $ch^+(X_i) \not\subseteq \boldsymbol{O}^X$.*

Theorem 3.1. *Let $\boldsymbol{O}^X := \textsc{FindOx}(\mathcal{G}, \boldsymbol{X}, Y)$, and $\boldsymbol{X}' := \boldsymbol{O}^X \cap \boldsymbol{X}$. Taking $\boldsymbol{Z}$ as the Markov Boundary of $\boldsymbol{O}^X$ in $\mathcal{G}^Y_{\underline{X'}}$ and $\boldsymbol{X}^B$ as the boundary actions of $\boldsymbol{O}^X$, the sets $\boldsymbol{Z}_i = (\boldsymbol{Z} \cup \boldsymbol{X}^B) \cap \text{before}(X'_i)$ for each $X'_i \in \boldsymbol{X}'$ are a valid sequential π-backdoor for $(\mathcal{G}, \boldsymbol{X}', Y)$.*

Theorem 3.2. *Let $\boldsymbol{O}^X := \textsc{FindOx}(\mathcal{G}, \boldsymbol{X}, Y)$. Suppose that there exists a sequential π-backdoor for $\boldsymbol{X}" \subseteq \boldsymbol{X}$. Then $\boldsymbol{X}" \subseteq \boldsymbol{O}^X$.*

Combined together, the above theorems show that FindOx finds the *maximal* subset of $\boldsymbol{X}$ where a sequential π-backdoor exists (Thm. 3.2,Thm. 3.1), and can be constructed through the application of a Markov Boundary over $\boldsymbol{O}^X$ (Thm. 3.1), which verifies that FindOx is both necessary and sufficient for generating a valid sequential π-backdoor:

Theorem 3.3. *Let $\boldsymbol{O}^X$ be the output of $\textsc{FindOx}(\mathcal{G}, \boldsymbol{X}, Y)$. A sequential π-backdoor exists for $(\mathcal{G}, \boldsymbol{X}, Y)$ if and only if $\boldsymbol{X} \subseteq \boldsymbol{O}^X$.*

We exemplify the use of Alg. 1 through the example in Fig. 2c. Considering the temporal order $\{X_1, Z, W, X_2, Y\}$, the algorithm starts at Y, which has no children and is not an element of $\boldsymbol{X}$, so is not added to $\mathscr{O}^X$. It then carries on to X_2, which is checked for a valid conditioning set. Here, the subgraph of the c-component of X_2 is simply $W \rightarrow X_2$, with no elements in the c-component other than W, and therefore we have $\mathscr{O}^X = \{X_2 : X_2\}$. Next, W has X_2 as its child, which maps to X_2 in $\mathscr{O}^X$. Once again, there are no other elements in W's c-component, so $\mathscr{O}^X = \{X_2 : X_2, W : X_2\}$. Since Z doesn't have its children in the keys of $\mathscr{O}^X$, and is not an element of $\boldsymbol{X}$, it is skipped, leaving only X_1. Since X_1's children (W) are in $\mathscr{O}^X$, we check conditioning using X_2. This time, we have $X_1 \leftrightarrow Z$ as the c-component subgraph, and Z comes before X_2, which satisfies the check. Both X_1 and X_2 are in the keys of $\mathscr{O}^X$, for which the Markov Boundary in $\mathcal{G}^Y_{\underline{\boldsymbol{X}}}$ is $\{Z\}$, and the boundary actions are $\{X_2\}$. This results in the sets $\boldsymbol{Z}_1 = \emptyset$ and $\boldsymbol{Z}_2 = \{Z\}$, which are a valid sequential π-backdoor.

When run on Fig. 2d, the algorithm tests that $(W \not\perp\!\!\!\perp Y)$ in $W \leftrightarrow Y$, so W can't be in $\boldsymbol{O}^X$. This means that Z won't be in $\mathscr{O}^X$ and therefore X_1 must have Z before it in temporal order, otherwise $Z \leftrightarrow X_1$ will have $(X_1 \not\perp\!\!\!\perp Z)$ rather than $(X_1 \perp\!\!\!\perp Z|Z)$ when checking conditioning. Finally, in Fig. 1d, the algorithm recognizes that X_1 cannot be part of any valid imitator, and returns $\boldsymbol{O}^X = \{X_2\}$, meaning that X_1 must still be controlled by the expert, while X_2 can be left to the imitator.

4 Necessity of Sequential π-Backdoor for Imitation

In this section, we show that the sequential π-backdoor is *necessary* for imitability, meaning that the sequential π-backdoor is complete.

A given imitation problem can have multiple possible conditioning sets satisfying the sequential π-backdoor, and a violation of the criterion for one set does not preclude the existence of another that satisfies the criterion. We therefore use the output of the algorithm, which returns a unique set $\boldsymbol{O}^X$ for each problem to prove the necessity of the sequential π-backdoor:

Theorem 4.1. *Let $\boldsymbol{O}^X := \textsc{FindOx}(\mathcal{G}, \boldsymbol{X}, Y)$. Suppose $X_i \in \boldsymbol{X} \setminus \boldsymbol{O}^X$. Then $\boldsymbol{X}$ is not imitable with respect to Y in $\mathcal{G}$.*

Theorem 4.2. *If there do not exist conditioning sets satisfying the sequential π-backdoor criterion for $(\mathcal{G}, \boldsymbol{X}, Y)$, then $\boldsymbol{X}$ is not imitable with respect to Y in $\mathcal{G}$.*

The proof of Thm. 4.1 relies on the construction of an adversarial SCM for which Y can detect the imitator's lack of access to the latent variables. For example, in Fig. 2a, Z can carry information about the latent variable U to Y, and is only determined after the decision for the value of X is made. Setting $U \sim Bern(0.5), X := U, Z := U, Y := X \oplus Z$ leaves the imitator with a performance of $\mathbf{E}[\hat{Y}] = 0.5$.

Another example with similar mechanics can be seen in Fig. 2c. If the variables are determined in the order (X_1, W, X_2, Z, Y), then the problem is not imitable, since Z can transfer information about the latent variable U to Y, while X_2 has no way of gaining information about U, because the action at X needed to be taken without context.

Finally, observe Fig. 2d. If Z is determined *after* X_1, the imitator must guess a value for X_1 without this side information, which is then combined with U_2 at W. An adversary can exploit this to construct a distribution where guessing wrong can be detected at Y as follows: $U_1 \sim Bern(0.5)$, $Z, X := U_1, U_2 \sim (Bern(0.5), Bern(0.5))$ (that is, U_2 is a tuple of two binary variables, or a single variable with a uniform domain of $0, 1, 2, 3$). Then setting $W = U_2[Z]$ ([] represents array access, meaning first element of tuple if $Z = 0$ and second if $Z = 1$), and $X_2 := W, Y := (U_2[X_1] == X_2)$ gives $\mathbf{E}[Y] = 1$ only if π_{X_1} guesses the value of U_1, meaning that the imitator can never achieve the expert's performance. This construction also demonstrates non-imitability when X_1 and Z are switched (Fig. 2c with $W \leftrightarrow Y$ added, and X_1 coming before Z in temporal order).

Due to these results, after running Alg. 1 on the domain's causal structure, the imitator gets two pieces of information:

1. Is the problem imitable? In other words, is it possible to use only observable context variables, and still get provably optimal imitation, despite the expert and imitator having different information?
2. If so, what context should be included in each action? Including/removing certain observed covariates in an estimation procedure can lead to different conclusions/actions, only one of which is correct (known as "Simpson's Paradox" in the statistics literature (Pearl, 2000)). Furthermore, when performing actions sequentially, some actions might not be imitable themselves (X_1 in Fig. 2c if Z after X_1), which leads to bias in observed descendants (W) - the correct context takes this into account, using only covariates known not to be affected by incorrectly guessed actions.

The result can then be used as input to existing behavioral cloning and inverse RL algorithms, guaranteeing an unbiased result.

5 Experiments

We performed 2 experiments (for full details, refer to Appendix B), comparing the performance of 4 separate approaches to determining which variables to include in an imitating policy:

1. **All Observed (AO)** - Take into account all variables available to the imitator at the time of each action. This is the approach most commonly used in the literature.
2. **Observed Parents (OP)** - The expert used a set of variables to take an action - use the subset of these that are available to the imitator.
3. **π-Backdoor** - In certain cases, each individual action can be imitated independently - allowing usage of a single-action imitation criterion.
4. **Sequential π-Backdoor (ours)** - The method developed in this paper, which takes into account multiple actions in sequence.

The first experiment consists of running behavioral cloning on simulations of randomly sampled distributions consistent with a series of causal graphs designed to showcase common situations. For each causal graph, 10,000 random discrete causal models were sampled, representing the environment as well as expert performance, and then the expert's policy $\boldsymbol{X}$ was replaced with imitating policies approximating $\pi(X_i) = P(X_i|ctx(X_i))$, with context ctx determined by each of the 4 tested methods in turn. Our results are shown in Table 1, with causal graphs shown in the first column, temporal ordering of variables in the second column, and absolute distance between expert and imitator for the 4 methods in the remaining columns. In the first row, including Z when developing a policy for $\boldsymbol{X}$ leads to a biased answer, which makes the average error of using all observed covariates (red) larger than just the sampling fluctuations present in the other columns. Similarly, Z needs to be taken into account in row 2, but it is not explicitly used by $\boldsymbol{X}$, so a method relying only on observed parents leads to bias here. In the next row, Z is not observed at the time of action X_1, making the π-backdoor incorrectly claim non-imitability. Our method recognizes that X_2's policy can fix the error made at X_1, and is the only method that leads to an unbiased result. Finally, in the 4th row, the non-causal approaches have no way to determine non-imitability, and return biased results in all such cases.

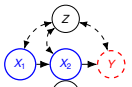
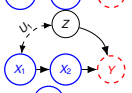
#	Structure	Order	**Seq. π-Backdoor**	π-Backdoor	Observed Parents	All Observed
1		Z, X_1, X_2, Y	$0.04 \pm 0.04\%$	$0.04 \pm 0.03\%$	$0.05 \pm 0.04\%$	$\mathbf{0.13 \pm 0.18\%}$
2		$\boldsymbol{Z, X_1, X_2, Y}$	$0.05 \pm 0.03\%$	$0.05 \pm 0.03\%$	$\mathbf{0.20 \pm 0.25\%}$	$0.05 \pm 0.03\%$
3		$\boldsymbol{X_1, Z, X_2, Y}$	$0.04 \pm 0.03\%$	**Not Imitable**	$\mathbf{0.27 \pm 0.40\%}$	$\mathbf{0.26 \pm 0.39\%}$
4		X_1, Z, X_2, Y	Not Imitable	Not Imitable	$\mathbf{0.19 \pm 0.29\%}$	$\mathbf{0.19 \pm 0.29\%}$

Table 1: Values of $|Y - \hat{Y}|$ from behavioral cloning using different contexts in randomly sampled models consistent with each causal graph.

The second experiment used continuous highway vehicle trajectory data as measured by drone from the HighD dataset (Krajewski et al., 2018), enriched with synthetic causal structure. A neural network was trained for each action-policy pair using standard supervised learning approaches, leading to the results shown in Fig. 4. The causal structure was not imitable from the single-action setting, so the remaining 3 methods were compared to the optimal reward, showing that our method approaches the performance of the expert, whereas non-causal methods lead to biased results. Full details of model construction, including the full causal graph are given in Appendix B.2.

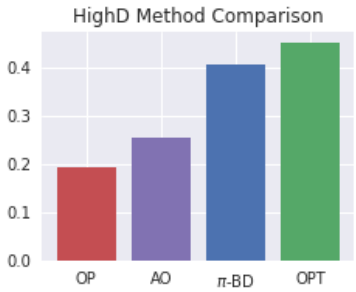


Figure 4: Results of applying standard supervised learning techniques to causally-enhanced HighD data with different sets of variables as input at each action. OPT is the ground truth expert's performance, π-BD represents our method, AO is all observed, and OP represents observed parents.

6 Limitations & Societal Impact

There are two main limitations to our approach: (1) Our method focuses on the causal diagram, requiring the imitator to provide the causal structure of its environment. This is a fundamental requirement: raw observations alone are provably insufficient to make claims about the effects of actions. Any agent wishing to operate in environments with latent variables must somehow encode the additional knowledge required to make such inferences from observations. (2) Our criterion only takes into consideration the causal structure, and not the associated data $P(\boldsymbol{o})$. Data-dependent methods can be computationally intensive, often requiring density estimation. If our approach returns "imitable", then the resulting policies are guaranteed to give perfect imitation, without needing to process large datasets to determine imitability.

Finally, advances in technology towards improving imitation can easily be transferred to methods used for impersonation - our method provides conditions under which an imposter (imitator) can fool a target (Y) into believing they are interacting with a known party (expert). Our method shows when it is provably impossible to mitigate an impersonation attack. On the other hand, our results can be used to ensure that the causal structure of a domain cannot be imitated, helping mitigate such issues.

7 Conclusion

Great care needs to be taken in choosing which covariates to include when determining a policy for imitating an expert demonstrator when expert and imitator have different views of the world. The wrong set of variables can lead to biased, or even outright incorrect predictions. Our work provides general and complete results for the graphical conditions under which behavioral cloning is possible, and provides an agent with the tools needed to determine the variables relevant to its policy.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. Survey: Robot programming by demonstration. *Handbook of robotics*, 59(BOOK_CHAP), 2008.

Etesami, J. and Geiger, P. Causal transfer for imitation learning and decision making under sensor-shift. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, New York, NY, 2020. AAAI Press.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

Krajewski, R., Bock, J., Kloeker, L., and Eckstein, L. 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)*, pp. 2118–2125, 2018. doi: 10.1109/ITSC.2018.8569552.

Mahler, J. and Goldberg, K. Learning deep policies for robot bin picking by simulating robust grasping sequences. In *Conference on robot learning*, pp. 515–524, 2017.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv:1312.5602 [cs]*, December 2013.

Muller, U., Ben, J., Cosatto, E., Flepp, B., and Cun, Y. L. Off-road obstacle avoidance through end-to-end learning. In *Advances in neural information processing systems*, pp. 739–746, 2006.

Mülling, K., Kober, J., Kroemer, O., and Peters, J. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32(3):263–279, 2013.

Ng, A. Y., Russell, S. J., et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, pp. 663–670, 2000.

Osa, T., Pajarinen, J., Neumann, G., Bagnell, J. A., Abbeel, P., Peters, J., et al. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics*, 7(1-2):1–179, 2018.

Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Pearl, J. *Causality: Models, Reasoning and Inference*. 2000.

Pearl, J. and Robins, J. M. Probabilistic Evaluation of Sequential Plans from Causal Models with Hidden Variables. *arXiv:1302.4977 [cs]*, 1995.

Pomerleau, D. A. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pp. 305–313, 1989.

Rizzolatti, G. and Craighero, L. The mirror-neuron system. *Annu. Rev. Neurosci.*, 27:169–192, 2004.

Sutton, R. S., Barto, A. G., et al. *Reinforcement Learning: An Introduction*. MIT press, 1998.

Syed, U. and Schapire, R. E. A game-theoretic approach to apprenticeship learning. In *Advances in neural information processing systems*, pp. 1449–1456, 2008.

Tian, J. *Studies in Causal Reasoning and Learning*. PhD thesis, Computer Science Department, University of California, Los Angeles, CA, November 2002.

Tian, J. and Paz, A. Finding Minimal D-separators. pp. 15, 1998.

Tian, J. and Pearl, J. A General Identification Condition for Causal Effects. pp. 7, 2002.

van der Zander, B. and Liśkiewicz, M. Finding minimal d-separators in linear time and applications. In *Uncertainty in Artificial Intelligence*, pp. 637–647. PMLR, 2020.

WIDROW, B. Pattern-recognizing control systems. *Compurter and Information Sciences*, 1964.

Zhang, J., Kumor, D., and Bareinboim, E. Causal Imitation Learning with Unobserved Confounders. pp. 27, 2020.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

8 Paper Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   [Yes] There is an additional discussion of limitations in sec. 6
   (b) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See sec. 6
   (d) Did you describe the limitations of your work? [Yes] See sec. 6
2. If you are including theoretical results
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are available in the supplementary material
3. If you ran experiments
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] Our code is not being included in the submission, however, we can release code upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Available in the appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our experiments include confidence intervals where applicable
   (d) Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] Our experiments did not require significant computational resources
4. If you are using existing assets
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the creators of the HighD dataset
   (b) Did you mention the license of the assets? [Yes] The license of the assets is available in the appendix
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] This is a commercial dataset.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] Our dataset only contains anonymous trajectories over short strips of highway
5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]