Learning Human Driving Behaviors with Sequential Causal Imitation Learning

Kangrui Ruan, Xuan Di
Columbia University, New York, NY, USA
kr2910@columbia.edu, sharon.di@columbia.edu

Abstract
Learning human driving behaviors is an efficient approach for self-driving vehicles. Traditional Imitation Learning (IL) methods assume that the expert demonstrations follow Markov Decision Processes (MDPs). However, in reality, this assumption does not always hold true. Spurious correlation may exist through the paths of historical variables because of the existence of unobserved confounders. Accounting for the latent causal relationships from unobserved variables to outcomes, this paper proposes Sequential Causal Imitation Learning (SeqCIL) for imitating driver behaviors. We develop a sequential causal template that generalizes the default MDP settings to one with Unobserved Confounders (MDPUC-HD). Then we develop a sufficient graphical criterion to determine when ignoring causality leads to poor performances in MDPUC-HD. Through the framework of Adversarial Imitation Learning, we develop a procedure to imitate the expert policy by blocking π-backdoor paths at each time step. Our methods are evaluated on a synthetic dataset and a real-world highway driving dataset, both demonstrating that the proposed procedure significantly outperforms non-causal imitation learning methods.

Introduction
Imitation Learning (IL) presents a promising paradigm for autonomous driving (Pomerleau 1988; Bojarski et al. 2016, 2017; Bansal, Krizhevsky, and Ogale 2018; Codevilla et al. 2019). There are two major types in IL: behavioral cloning (BC) and inverse reinforcement learning (IRL). BC methods directly learn an approximate conditional distribution from any given state to the expert’s action. Instead, IRL learns the implicit reward function, which is optimal to the expert behaviors; then using this learned reward function, RL methods are employed to obtain a policy. Inspired by Generative Adversarial Networks (GANs) and IRL (Goodfellow et al. 2014; Gulrajani et al. 2017), the framework of Adversarial Imitation Learning (AIL) is proposed (Ho and Ermon 2016; Li, Song, and Ermon 2017; Fu, Luo, and Levine 2017). In AIL, the policy is trained by producing expert-like state-action pairs to fool the discriminator. While training, the discriminator can provide a reward signal to help the policy proceed to expert-like zones.

However, IL methods rely on the assumption that all expert features are fully observed and recorded. When there exist unobserved variables inside the expert’s demonstrations, critical problems arise. Causal Inference (CI) addresses those issues by investigating the causal relationships among observed and unobserved variables (Spirtes et al. 2000; Pearl 2009; Peters, Janzing, and Schölkopf 2017). There has been some research focused on the combination of IL and Causality. Causal Imitation Learning (CIL) was recently proposed by (Zhang, Kumar, and Barenboim 2020), which focuses on learning a policy within the limit of non-sequential one-stage settings. The authors of (Etesami and Geiger 2020) study the causal transfer problem by assuming that the relationships among variables are linear. The paper (de Haan, Jayaraman, and Levine 2019) ignores unobserved confounders and assumes the reward and the expert can be easily accessible.

So far we found one paper (Kumor, Zhang, and Barenboim 2021) related to SeqCIL, and the major differences between their research and ours are listed here. Their diagrams are within finite-stage settings, but we work in the γ-discounted infinite horizon setting. Their experiments are mainly based on synthetic datasets with simplified reward functions. However, our experiments include a realistic dataset whose actual reward function cannot even be accessed. Their method mainly focuses on behavioral cloning, while we also show the internal connection between AIL and SeqCIL.

Little research has been conducted for sequential imitation learning when unobserved variables exist. To demonstrate the issue of ignoring unobserved variables resulting in an unacceptable policy, we begin with an introductory example.

Example 1. Suppose a scenario when an imitator needs to learn how to correctly accelerate $A_t$. $S_t$ denotes the velocity and locations of the ego vehicle and the surrounding vehicles. In reality, there exist some unobserved variables inside the expert demonstrations. Human drivers utilize the vehicle light information $U^{(L)}_t$, i.e., the tail light from the front vehicle 1 and the turn signal from the left vehicle 2. However, $U^{(L)}_t$ is not properly recorded, which is common in some real-world datasets, such as NGSIM (Alexiadis et al. 2004) or highD (Krajewski et al. 2018). When the demon-
strator drove, the road can be slippery $U^1_{t}(S,R)$, which is unknown to the imitator. The level of the expert driving skills $U_π$ is ignored. $H_{t-1}$ encodes the history information, which the imitator does not take into account. The imitator only takes $S_t$ as input, while the expert makes a decision based on $U_π, H_{t-1}, S_t, U^L_t$. The latent reward is evaluated based on $S_t, A_t, U^1_{t}(S,R)$. Fig. 1 depicts this situation and its graphical representation.

Consider a numerical instance where variables $A_t, S_t, R_t, H_{t-1}, U^L_t, U^1_{t}(S,R), U_π ∈ \{0, 1\}$; their values are generated based on the functions: $U_π ← 0$, $S_t ← H_{t-1} ⊕ U^1_{t}(S,R), U^L_t ← S_t$, the expert policy $π_E : A_t ← U_π ⊕ H_{t-1} ⊕ S_t ⊕ U^L_t$, the implicit reward $R_t ← ¬(A_t ⊕ U^1_{t}(S,R) ⊕ S_t)$; variables $H_{t-1}, U^1_{t}(S,R)$ are uniformly sampled over $\{0, 1\}$; the operator $⊕$ means exclusive-or.

However, when ignoring the unobserved variables, the imitator under the same demonstrator drove, the road can be slippery $U^1_{t}(S,R)$, which is ignored. $H_{t-1}$ encodes the history information, which the imitator does not take into account. The imitator only takes $S_t$ as input, while the expert makes a decision based on $U_π, H_{t-1}, S_t, U^L_t$. The latent reward is evaluated based on $S_t, A_t, U^1_{t}(S,R)$.

The rest of this paper is organized as follows: Section exhibits the preliminaries and formally defines the problem of SeqCIL. Section presents the proposed causal template for human driving behaviors, i.e., MDPUC-HD. Based on this template, Section explicates why unobserved variables cannot be ignored, and provides the sequential causal imitator to address such issues. We propose to solve SeqCIL through the framework of AIL in Section, and experiments are conducted in Section.

Sequential Causal Imitation Learning

Conventions: In this paper, capitalized letters represent random variables, e.g. $R$ and lowercase letters are their specific values, e.g. $r$; the probability distribution of a random variable is capital $P(R)$, and $p(r)$ is for the mass at value $R = r$. $S$ denotes a set. $D_{KL}$ represents the Kullback–Leibler (KL) divergence between two distributions. Without explicit remark, all proofs are shown in the appendix of this paper.

Preliminaries on Imitation Learning

Imitation Learning: $S$ represents the state space, and $A$ represents the action space. The expert state space $S_π$ and the chosen imitator state space $S_π$ can be different. The policy space $Π$ is the set of all stationary stochastic policies that take actions in $A$ given states in $S_π$, denoted by $Π : S_π → A$.

Occupancy measure: Under the policy $π$, the state-action occupancy measure $ρ_π(s,a) : S_π × A → R$ is defined as $ρ_π(s,a) = P(s,a|do(π)) = P(s,a;π) = ρ_π(s)π(a|s)$, where $ρ_π(s)$ denotes the unnormalized discounted future state distribution when following the policy $π$: $ρ_π(s) = ∑_{t=0}^{∞} γ^t P_π(st = s)$.

Expert Regime: $π_E$ always refers to the expert policy. $P π_E(·)$ denotes the distribution for any feasible random variable under the expert $π_E$ regime. $P(·)$ denotes the observational distribution (“demonstrations”). In this paper, demonstrations are generated by the expert, so that $P π_E(·) = P(·)$.

Imitator Regime: $P π(·)$ denotes the distribution under the imitator $π$ regime. Generally, $P π(·) ≠ P(·)$, except the situations when some graphical conditions are satisfied. (More details can be found in Appendix Sec. A)

Adversarial Imitation Learning (AIL): AIL learns a policy by matching occupancy measures between the expert and the imitator, and its training procedure includes updating the discriminator and the policy simultaneously. Generally, $S_π$ is manually chosen. People usually choose $S_π$ the same as $S_π$, but they can be different.

To illustrate the idea of “regime”, we take $ρ π_1(s)$ and $ρ π_2(s)$ as an example. $ρ π_1(s)$ is under policy $π_1$ regime. $ρ π_2(s)$ is under policy $π_2$ regime. We distinguish different regimes, because different policies generally lead to different state distributions, i.e., $ρ π_1(s) ≠ ρ π_2(s)$. (More details are in Appendix Sec. A.)
Causal Template for Human Driving Behaviors

In this paper, we propose a causal template: Markov Decision Processes with Unobserved Confounders for Human Driving (MDPUC-HD), depicted in Fig. 2.

When using real-world driving datasets, it is inappropriate to assume that expert covariates are fully observed. There are lots of unobserved variables in demonstrations. Similar to Example 1, the light information $U_t^{L}$ is not appropriately recorded in some real-world driving datasets. $U_z$ represents the internal latent code, e.g., levels of driving skills and inconsistent policies. Additionally, human drivers usually take actions based on previous actions, suggesting a cause-effect relationship between $A_{t-1}$ and $A_t$ (de Haan, Jayaraman, and Levine 2019; Codevilla et al. 2019). $U_t^{A}$ denotes those unknown actions which affect the environment dynamics, e.g., the actions from other vehicles. $U_t^{(S,R)}$ are the unobserved variables that affect both the reward and the state. The expert policy $\pi_E$ denotes the condition distribution which is unknown to the imitator. The implicit reward $R_t$ is decided by $S_t, A_t, U_t^{(S,R)}$. The observability of each mentioned variable is summarized in Table 1.

<table>
<thead>
<tr>
<th>Variables</th>
<th>Expert</th>
<th>Imitator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_t^{x}$</td>
<td>known</td>
<td>unknown</td>
</tr>
<tr>
<td>$U_t^{L}$</td>
<td>known</td>
<td>unknown</td>
</tr>
<tr>
<td>$U_t^{(S,R)}$</td>
<td>known</td>
<td>unknown</td>
</tr>
<tr>
<td>$U_t^{A}$</td>
<td>unknown</td>
<td>unknown</td>
</tr>
<tr>
<td>$R_t$</td>
<td>known</td>
<td>unknown</td>
</tr>
</tbody>
</table>

Table 1: The observability of variables for the expert and the imitator

Compared to an MDP, an MDPUC-HD is more general and realistic to practical applications: (1) An MDPUC-HD does not assume that the expert covariates are fully observed, because of $U_t^{L}$ and $U_t^{x}$; (2) In an MDPUC-HD, $R_t$ can depend on variables not inside $S_t$, i.e., $U_t^{(S,R)}$; (3) The transition dynamics may not be successfully recovered, because of the unobserved variables $U_t^{A}$ and $U_t^{(S,R)}$; (4) Following a MDPUC-HD, the dataset does not need to satisfy the constraint $A_t \perp \perp (S_{t-1}, A_{t-1}) | S_t$. While following MDPs, the dataset should meet this constraint. In summary, the proposed causal template is a generalized version of MDPs, which imposes fewer constraints.

MDPUC-HDs are orthogonal and complementary to POMDPs with different focuses (Zhang and Bareinboim 2021). MDPUC-HDs concentrate on the causal relationships. It is possible that unobserved confounders still exist in POMDPs. In MDPUC-HDs, not all expert inputs are observed, e.g., $U_t^{L}$. In POMDPs, the actual states are hidden, and all decisions are made according to the entire history, i.e., POMDPs are non-Markovian. However, the cru-
Causal Markov Property

**Proposition 1** (Causal Markov Property\(^3\)). Following the causal diagram \(\mathcal{G}\) in Fig. 2, the following statements hold:

1. Given the current state \(S_t\) and action \(A_t\), the transition to the next state \(S_{t+1}\) is independent of the past history:
   \[
P(S_{t+1}|S_t, A_t, S_{t-1}, A_{t-1}, \ldots, S_0, A_0) = P(S_{t+1}|S_t, A_t) \tag{2}
   \]

2. Given the 2-step information tuple \((S_t, S_{t-1}, A_t, A_{t-1})\), the reward \(R_t\) is independent of the past history:
   \[
P(R_t|S_t, S_{t-1}, A_t, A_{t-1}, \ldots, S_0, A_0) = P(R_t|S_t, A_t, S_{t-1}, A_{t-1}) \tag{3}
   \]

In general, Prop. 1 is valid for a series of graphs:

**Proposition 2.** Given the causal diagram \(\mathcal{G}\) following Fig. 2, removing any bi-directed arrow or unobserved variable, Causal Markov Property (Prop. 1) is still tenable.

Prop. 2 is intuitive: after removing any bi-directed arrow or unobserved variable, the new graph still satisfies the original (conditional) independence constraints.

**Sequential Causal Imitator**

With Prop. 1 and 2 defined above, in this section, we first explain why the non-causal imitator fails, and then propose a solution using the sequential causal imitator.

---

\(^3\)All strict proofs are provided in Appendix.

---

**Why Non-causal Imitator Fails**

To analyze the failure reason for the non-causal imitator, we begin with Example 2, which shares the similar driving setting as Example 1.

**Example 2.** Consider an instance where demonstrations follow Fig. 2. When the history information \(\{S_t, A_t, U^{(A)}_t, U^{(L)}_t, U^{(S,R)}_t\}_{t=0}^{T-1}\) is summarized as \(H_{t-1}\), the right part in Fig. 1 encodes the similar causal relationships among \(S_t, A_t, U^{(L)}_t, R_t\) as in Fig. 2.

Variables \(A_t, S_t, R_t, U^{(S,R)}_t, U^{(A)}_t, U^{(L)}_t \in \{0, 1\}\); their values are generated by processes: \(U_\pi \sim \text{Bern}(0.5)\), \(U^{(L)}_t \leftarrow S_t, U^{(A)}_t \leftarrow S_t; \text{at } t = 0, S_0 \leftarrow U^{(S,R)}_0\). Later, when \(t \geq 1\), \(S_t\) is determined by the function: \(S_t \leftarrow U^{(A)}_{t-1} \oplus S_{t-1} \oplus A_{t-1} \oplus U^{(S,R)}_t\). The reward \(R_t\) is defined as \(R_t \leftarrow \neg(S_t \oplus A_t \oplus U^{(S,R)}_t)\). The implicit expert policy \(\pi_E\) is defined as:

\[
A_t \leftarrow \begin{cases} S_0 \oplus U^{(L)}_i \oplus U_{\pi}, & \text{if } t = 0 \\ A_{t-1} \oplus S_t \oplus U^{(L)}_t \oplus U_{\pi}, & \text{otherwise} \end{cases} \tag{4}
\]

; the operator \(\oplus\) means exclusive-or. Non-causal imitator \(\pi(a_t|s_t)\), is only capable of achieving sub-optimal performance.

When unobserved confounders exist, as Example 2 demonstrates, the non-causal imitator \(\pi(a_t|s_t)\) fails to recover a good policy. The underlying reason is related to the \(\pi\)-backdoor paths between \(R_t\) and \(A_t\): (1) \(A_t \leftarrow S_t \rightarrow R_t\). (2) \(A_t \leftarrow A_{t-1} \rightarrow S_t \leftarrow R_t\), where \(S_t\) is a collider. When \(S_t\) is conditioned, path (1) is blocked but the colliding path (2) is open, so that some spurious correlation is learned. Formally, we prove a generalized theorem to analyze the criterion when the non-causal imitator fails:

---

4586
**Theorem 1** (Not Imitable in the Same C-Component). Given a causal diagram $G$ and a policy space $\Pi$, at time step $t$, if $\pi_E$ and $\pi_t$ are in the same C-Component of $\pi_t$'s ancestral graph, then there exist two models $M_1, M_2$, which have the same distribution of observed variables, but there exists no policy $\pi \in \Pi$ such that $P(R_t; M_1) = P(R_t; do(\pi); M_1)$ and $P(R_t; M_2) = P(R_t; do(\pi); M_2)$, i.e., $P(R_t)$ is not imitable.

C-Component (Tian 2002) is a set of nodes that are connected by bi-directed arrows. Here we use a fake node $X$ as an example. A directed path from a node $X$ to $R_t$ is a path composed of directed edges ($X \to \cdots \to R_t$). If there is a directed path from a node $X$ to $R_t$ or $X = R_t$, then this node $X$ is an ancestor of $R_t$. $R_t$'s ancestral graph is composed of nodes that are ancestors of $R_t$. To illustrate the idea of the same C-Components of $R_t'$s ancestral graph, take $S_t$ as example. $S_t$ is connected with $R_t$ by bi-directed arrows, and there is a directed path from $S_t$ to $R_t$.

In Example 1, $\pi_E$ and $S_t$ are $R_t'$s ancestors because of paths $\pi_E \to A_t \to R_t$ and $S_t \to R_t$. The path $\pi_E \leftrightarrow S_t$ makes $\pi_E$ and $R_t$ in the same C-Components. Therefore, $\pi_E$ and $R_t$ are in the same C-Component of $R_t$'s ancestral graph, and the non-causal imitator $\pi(a_t|s_t)$ fails. Intuitively, Thm. 1 also suggests that more variables should be observed, so that less confounding effect exists. This result can causally justify why more sensors are required to obtain a better policy, when recording human driving demonstrations (Jeyachandran 2020).

**Sequential Causal Imitator**

In this section, we circumvent the above issue by exploiting topological graphical conditions.

**Definition 1** (Sequential Causal Imitator). Given a causal diagram $G$ following Fig. 2 and a policy space $\Pi$, $\pi(a_t|s_{t,\text{causal}})$ is called “sequential causal imitator” with respect to $(G, \Pi)$, if the set $S_{t,\text{causal}}$ blocks all $\pi$-backdoor paths between $R_t$ and $A_t$, i.e., $(R_t \perp \perp A_t|s_{t,\text{causal}})_{\text{G}_{\text{\Pi}}}$. Although there exist diverse sets that can block all $\pi$-backdoor paths between $R_t$ and $A_t$, we only consider the minimal set\(^4\). Later, we will show that a larger set does not necessarily lead to a better performance.

Below, two propositions show the warrant for the sequential causal imitator with a satisfactory performance in multiple graphs.

**Proposition 3.** Given the causal diagram $G$ following Fig. 2 and a policy space $\Pi$, the sequential causal imitator $\pi(a_t|s_{t,\text{causal}})$ is guaranteed to match the performance of the expert, $p_{\pi_t}(r_t) = p(r_t)$, at time step $t$. (The strict proof is in Appendix.)

Prop. 3 shows: without knowing the expert policy $\pi_E$, the sequential causal imitator $\pi(a_t|s_{t,\text{causal}})$ can still achieve $p_{\pi_t}(r_t) = p(r_t)$. Prop. 4 generalizes it to graphs with fewer bi-directed arrows or unobserved variables.

**Proposition 4.** Given the causal diagram $G$ following Fig. 2, removing any bi-directed arrow or unobserved variable, Prop. 3 is still tenable.

Consider Example 2 again, denote the set $\{S_t, A_{t-1}, S_{t-1}\}$ by the set $S_{t,\text{causal}}$, which blocks all $\pi$-backdoor paths between $R_t$ and $A_t$. The causal relationships among $S_t, A_{t-1}, S_{t-1}$ are depicted in Fig. 3. With Prop. 3, the sequential causal imitator $\pi(a_t|s_{t,\text{causal}})$ is able to obtain the optimal expert performance $E[R_t; do(\pi)] = 1$, for $t \geq 1$.

**Adversarial Imitation Learning**

To obtain the sequential causal imitator, we propose to use the framework of AIL. Allowing the agent to explore the environment, AIL addresses the issue of “covariate shift” (Ross, Gordon, and Bagnell 2011). Apart from that, AIL facilitates the developed SeqCIL to explicitly match the performances of the expert and the imitator. To demonstrate how to leverage AIL for SeqCIL, we will implement GAIL below.

$f_r$ represents the implicit reward function, which takes $s_t, a_t, u_t^{(S,R)}$ as input. Although $u_t^{(S,R)}$ affects the true reward $R_t$ and the reward function $f_r$ is unknown, a surrogate reward signal can still be studied from demonstrations. With $S_{t,\text{causal}}$ equal to $\{S_t, A_{t-1}, S_{t-1}\}$, Prop. 1 implies $R_t \perp \perp S_{t-2}, A_{t-2}, \ldots, S_0, A_0|S_{t,\text{causal}}, A_t$. That is, given $(S_{t,\text{causal}}, A_t)$, the history $(S_{t-2}, A_{t-2}, \ldots, S_0, A_0)$ does not tell any newer information related to the reward $R_t$. This suggests that $(S_{t,\text{causal}}, A_t)$ can be treated as a proxy for the whole trajectory as the input to the discriminator ($D$).

We utilize the Wasserstein distance (Arjovsky, Chintala, and Bottou 2017) for occupancy measure matching between different regimes. When the discriminator cannot distinguish the causal state-action pairs coming from the imitator or the expert $\pi_E$, the occupancy measure is matched, as summarized in Prop. 5. The objective of the generator and the discriminator is given by:

$$\min_{\pi} \max_D E_{\pi_E} [D(s_{\text{causal}}, a)] - \lambda H(\pi)$$

where $\pi$ denotes the policy of the sequential causal imitator, $D$ is the discriminator taking causal state-action pair $(s_{\text{causal}}, a)$ as input. $H(\pi)$ is the $\gamma$-discounted causal entropy of the policy $\pi$. When updating the parameters of the policy, $\pi' = D(s_{\text{causal}}, a)$ plays like a proxy reward signal.

**Proposition 5.** When the discriminator $D$ cannot distinguish $(s_{\text{causal}}, a)$ generated from the sequential causal imitator $\pi$ or the expert $\pi_E$, $p_{\pi_t}(s_{\text{causal}}, a)$ matches $p_{\pi_E}(s_{\text{causal}}, a)$, and $\pi(a|s_{\text{causal}})$ matches $p_{\pi_E}(a|s_{\text{causal}})$. The convergence of the discriminator indicates the performance match, described in Eq. (1).

Under the imitator $\pi$ regime, the relationship between l.h.s. of Eq. (1) and the occupancy measure $\rho_\pi(s_{\text{causal}}, a)$ is given by:

\[4587\]
with Eq. (6), we will prove Prop. 5 in Appendix.

**Proximal Policy Optimization (PPO).** PPO is a model-free on-policy algorithm (Schulman et al. 2017). There are two principal variants: PPO-penalty and PPO-clip. PPO-penalty turns the trust-region constraint into a penalty $D_{KL, forward}(\pi_{old} \parallel \pi)$ to approximately restrict the size of the update. Instead, PPO-clip clips the probability ratio directly via:

$$
\mathbb{E} \left[ \sum_t \gamma^t R_t \mid \text{do}(\pi) \right] = \sum_{s_{causal}, a, u^{(S,R)}} p_\pi(s_{causal}, a, u^{(S,R)}) f_\tau(s, a, u^{(S,R)}) \\
= \sum_{s_{causal}, a, u^{(S,R)}} \rho_\tau(s_{causal}, a) p_\pi(u^{(S,R)}|s_{causal}, a) f_\tau(s, a, u^{(S,R)})
$$

(6)

where $\epsilon$ is a hyper-parameter to control the clip value, $\pi_{old}$ denotes the policy prior to the update, $A^{\pi_{old}}$ denotes the advantage function for the policy $\pi_{old}$.

Suggested by (Hsu, Mendler-Dünnér, and Hardt 2020), to better keep the new policy near the old policy and avoid common failure modes, our PPO objective is formulated as:

$$
\mathcal{L}_{PPO}(\theta) = \mathcal{L}_{CLIP}(\theta) - \beta_1 D_{KL, forward}(\pi_{old} \parallel \pi_\theta) - \beta_2 D_{KL, reverse}(\pi_\theta \parallel \pi_{old})
$$

(8)

where $\mathcal{L}_{CLIP}(\theta)$ is defined in Eq. (7) and $D_{KL}$ represents the KL divergence between two distributions, which is asymmetric. We use two soft constraints because generally $D_{KL, forward}(\pi_{old} \parallel \pi_\theta) \neq D_{KL, reverse}(\pi_\theta \parallel \pi_{old})$.

To summarize, Algorithm 1 reveals the training procedure of how to obtain a sequential causal imitator through GAIL. The convergence of the discriminator can be interpreted as the explicit signal that the performance is finally matched. Similar to the previous research (Ho and Ermon 2016; Fu, Luo, and Levine 2017), we implement the policy and the discriminator with Neural Networks and use Monte-Carlo methods to approximate expectations.

**Experiments**

We conduct experiments mainly on two classic driving datasets: a synthetic dataset simulating drivers’ car-following behaviors and a real-world highway driving dataset, Next Generation Simulation (NGSIM) (Alexiadis et al. 2004). Our experiments seek to answer the following questions: (1) Is the sequential causal imitator better than the non-causal imitator? (2) Is the proposed method robust to the real-world dataset, even without the ground truth reward function? (3) Will more temporal information, e.g., 3-step GAIL or recurrent policy, show a better result?
Synthetic Car-Following Dataset

Car-following is one of the most common and frequent scenarios in reality. The primary mission of car-following is to maintain safety, efficiency, and comfort by controlling the longitudinal dynamics, i.e., accelerating or braking.

State. In this experiment, the properties of the leading vehicle and ego vehicle are initialized randomly. At each time step $t$, the state $S_t$ is composed of the velocity of the ego vehicle, the velocity of the leading vehicle, and the gap between them, which is the same as the observation of the Intelligent Driver Model (IDM) (Treiber, Hennecke, and Helbing 2000).

Action. The action $A_t$ is the acceleration of the ego vehicle, which is continuous.

Reward. To simulate the complex real-world driving target, we need to design the reward by meeting the demands for safety, comfort, and efficiency. Safety is the major challenge. Therefore, we make use of time-to-collision (TTC), which measures the risk of collision if the speed difference of two vehicles is maintained (Hayward 1972). To quantify the level of human comfort when driving, we use jerk, the first derivative of acceleration (Bellem et al. 2018). For efficiency, to avoid driving too slow, the vehicles need to maintain a proper time headway. Finally, the reward $R_t$ is also dependent on $U_t^{(S,R)}$.

Unobserved Variables. $U_t^{(A)}$ denotes the acceleration of the leading vehicle, which affects the state transition but is unknown to both the expert and the imitator. $U_t^{(L)}$ represents the light tail indicator, when the leading vehicle brakes $U_t^{(L)} := 1$, otherwise 0. When the road is slippery, $U_t^{(S,R)} = 1$, only the TTC (safety) part of the reward is concerned, and the state $s_t$ is also affected.

Expert demonstrations are generated by running PPO-clip on the ground truth reward, which is similar to the settings of GAIL (Ho and Ermon 2016). During imitation learning, the environment reward is not provided anymore.

Performance Metrics: Similar to the settings in IL, we compare the average cumulative reward directly.

Results: According to Table 2, causal 2-step GAIL is much better than the well-established baselines.

<table>
<thead>
<tr>
<th>Models</th>
<th>Normalized Average Cumulative Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive BC</td>
<td>0.281</td>
</tr>
<tr>
<td>Non-causal GAIL</td>
<td>0.196</td>
</tr>
<tr>
<td>Causal 2-step GAIL</td>
<td><strong>0.892</strong></td>
</tr>
<tr>
<td>3-step GAIL</td>
<td>0.592</td>
</tr>
<tr>
<td>LSTM GAIL</td>
<td>0.238</td>
</tr>
</tbody>
</table>

Table 2: Normalized average cumulative reward for the car-following dataset. The result values are normalized so that 1.0 represents the performance of demonstrations.

When the action is negative, it is the deceleration.

Real-world NGSIM Dataset

The public NGSIM data records the real-world human driving trajectories for US Highway 101, and Interstate 80 Freeway (Alexiadis et al. 2004). Each dataset contains a total of 45 minutes of trajectories recorded per 0.1s, including precise information for location, speed, acceleration, surrounding vehicles. We focus on US Highway 101 here.

To offer a realistic driving environment, we adapt a Julia-based NGSIM simulator (Kuefler et al. 2017; Bhattacharyya et al. 2018). In the beginning, the environment is reset arbitrarily to choose a frame from NGSIM data. Inside this chosen frame, the ego-vehicle is randomly selected among all vehicles. Other traffic participants follow the replayed trajectory safely with proper extra braking mechanisms to avoid a potential collision with the ego vehicle.

State. At each time step $t$, the state $S_t$ consists of the following types of features extracted from the NGSIM data: (1) The features for the ego-vehicle, such as the width and the length, local lane curvature, the distance to the left and the right lane markers. (2) Temporal features, e.g., the time gap and time-to-collision (TTC). (3) Information of surrounding vehicles. Specifically speaking, multiple LiDAR beams are sent out by the ego vehicle and detect first struck objects with the information of the relative position and the range rate. (4) The information for the vehicle ahead of the fore vehicle. The simulator is done whenever the ego vehicle collides, drives off-road, or drives backward.

Action. In this experiment, longitudinal and lateral motions are based on the dynamics of a bicycle model. The action $A_t$ includes the acceleration and turn rate.

Performance Metrics: It is infeasible to directly compare the cumulative reward because we do not know the ground truth reward function when people drive. Additionally, there are infinitely many reward functions to make the expert demonstrations optimal, because the IRL problem is ill-posed (Nguyen, Low, and Jaillet 2015). Therefore, trajectories are used as a proxy to check. To characterize the difference, we make use of Root Mean Squared Error (RMSE), KL-divergence, and undesirable behaviors (Kuefler et al. 2017; Bhattacharyya et al. 2018).

RMSE. Suppose the imitator generates $m$ trajectories. Each predicted trajectory corresponds to one ground truth trajectory. $v$ is the variable of interest we want to compare. $v_{\pi,i}^{(i)}(t)$ denotes the simulated value executing the policy $\pi$ in the $i$th trajectory at time step $t$. $v_{\pi,E}^{(i)}$ denotes the true value generated by the expert. RMSE$(v)$ is defined below:

$$\text{RMSE}(v)_t = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (v_{\pi,E}^{(i)} - v_{\pi,i}^{(i)})^2} \quad (9)$$

We extract RMSE for the global position, lane offset, and speed over an entire time horizon up to 20 seconds.

KL divergence. KL divergence is to compare the dissimilarity for the predicted and actual actions distribution. A satisfied policy should share a similar distribution with expert demonstrations over the acceleration and turn rate.
Metrics on undesirable behaviors. Undesirable behaviors include off-road duration, collision rate, and hard braking rate, capturing the feasibility and reliability of the model. The collision rate is the ratio of trajectories when the ego vehicle collides with other vehicles. The hard brake rate calculates the situations when the deceleration is higher than a threshold. Off-road duration exhibits the average number of time steps when the moving ego vehicle is outside the road.

Results: Fig. 5 shows the results of RMSE. For each curve, the line corresponds to the mean value, and the shaded area represents the variance. Causal 2-step GAIL performs best among all models, while 3-step GAIL performs similarly in terms of the RMSE of lane offset. Table 3 shows the results of KL-divergence. Causal 2-step GAIL performs best for both motions, although 3-step GAIL also performs well for the turn rate. The results for undesired behaviors are shown in Fig. 4, demonstrating that causal 2-step GAIL is less likely to have undesired behaviors, and acts more like human beings.

To summarize, the sequential causal imitator learned through AIL performs the best with or without knowing the ground truth reward function, and more temporal information is not guaranteed to give us better results.

Conclusions
This paper exploited the qualitative knowledge for human drivers and proposed a causal template, MDPUC-HD, to model their driving behaviors. We explained why unobserved variables cannot be ignored for sequential demonstrations, and proposed a principled and general approach to solving SeqCIL with GAIL, with the potential to apply in other domains. The take-away message is: when unobserved variables exist, sequential causal imitators can better capture the expert behaviors than non-causal imitators.

Acknowledgments
This work is partially sponsored by the He Research Fund for Artificial Intelligence, Robotics and/or Autonomous Vehicles under Columbia’s SEAS Interdisciplinary Research Seed (SIRS).

References

4590


