3CIL: CAUSALITY-INSPIRED CONTRASTIVE CONDITIONAL IMITATION LEARNING FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation learning (IL) aims to recover an expert's strategy by performing supervised learning on the demonstration datasets. Incorporating IL in safety-crucial tasks like autonomous driving is promising as it requires less interaction with the actual environment than reinforcement learning approaches. However, the robustness of IL methods is often questioned, as phenomena like causal confusion occur frequently and hinder it from practical use. In this paper, we conduct causal reasoning to investigate the crucial requirements for the ideal imitation generalization performance. With insights derived from modeled causalities, we propose causality-inspired contrastive conditional imitation learning (3CIL), a conditional imitation learning method equipped with contrastive learning and action residual prediction tasks, regularizing the imitator in causal and anti-causal directions. To mitigate the divergence with experts in unfamiliar scenarios, 3CIL introduces a sample-weighting term that transforms the prediction error into an emphasis on critical samples. Extensive experiments in the CARLA simulator show the proposed method significantly improves the driving capabilities of models.

1 INTRODUCTION

Known as learning from demonstrations, imitation learning (IL) has attracted attention for its capability of replicating the behavior of experts in some tasks given only experts' demonstrations available. In many real-world applications, IL is widely used, because the desired behavior patterns are hard to construct, and the requirements for a competent model are difficult to summarize into a reward function or optimize objective. IL approaches can be divided into two categories, namely behavior cloning (BC) and inverse reinforcement learning (IRL). BC is one of the most prominent IL algorithms, which transforms the procedure of imitating experts' strategy into a simple supervised learning problem. For those tasks with strict safety concerns or expensive trial costs like autonomous driving, BC is preferred as provides an effective solution with no requirement on interactions with the real environment during training.

However, the performance of BC methods is often questionable, especially in complex environments. Many approaches (de Haan et al., 2019; Wen et al., 2020; Codevilla et al., 2019; Wen et al., 2022; Ortega et al., 2021) have investigated the factors that led to the problematic decision-making pattern of the imitator. What is common in these analyses is that they attribute the train-test performance gap to causal confusion: the reliance of the imitator on spurious correlations or shortcuts, instead of causal relations. Due to the lack of causal principles, models are prone to use features that are spurious correlated to the expert's actions, as relying on these correlations only needs fewer parameter updates to obtain a stable and low loss in the training phase. Moreover, the potential perception mismatch between the expert and the imitator may further impose the imitator's reliance on spurious correlations. Nevertheless, these correlations or shortcuts may hold only when the target environment has the same distribution as the distribution demonstrations sampled from. Therefore, imitators that are tempted by these shortcuts, perform poorly when testing them in new environments.

To alleviate the above problems of BC in visually complex tasks like autonomous driving, we incorporate the idea of causal reasoning to assist imitation. While typical techniques of causal reasoning

can not directly be deployed in these tasks with high-dimensional observations and partial visibility, modeling causal relations among concepts in the driving task can still provide indications for learning a robust imitator. By investigating the correspondence between causal relations and the imitator's behaviors, we identify crucial traits that a robust imitator must have. With supervisions from both causal direction and anti-causal direction, the latent state inferred by the imitator is urged to produce stable causal effects on its descendant node.

In this paper, we consider imitating experts' behavior to achieve autonomous driving, under the setting of conditional imitation learning (CIL) (Codevilla et al., 2018). Equipped with causality, we propose Causality-inspired Contrastive Conditional Imitation Learning (3CIL), an imitation learning method that incorporates contrastive learning and residual prediction tasks for better generalization. Our contributions are:

- Based on causal reasoning about the behavior cloning process in the driving task, we identify crucial traits that a robust imitator must have. By incorporating contrastive learning and action residual prediction objectives into imitation learning, we enhance the imitator's representation's robustness through influence from causal and anti-causal directions.
- We propose a sample-weighting process to emphasize scenarios that cause high divergences between the expert and imitator, guide the imitator to adapt to diverse situations.
- We conduct extensive experiments on the CARLA simulator (Dosovitskiy et al., 2017) to demonstrate the effectiveness of the proposed 3CIL approach and the relations between causal insights and actual performance.

2 Preliminaries and Definitions

2.1 Spurious correlations in behavior cloning

Compared to online reinforcement learning (RL) or IRL methods, agents trained with BC are more vulnerable to spurious correlations in data. As the imitator cannot interact with the target environment in the training phase, it can not test or validate its learned pattern but only count on offline evaluation metrics (e.g. frame-wise Mean Squared Error in steer angle prediction). Such phenomenon is known in the literature as causal confusion (de Haan et al., 2019): BC agents lack explicit causal understanding of their tasks.

Causal confusion becomes more evident in complex tasks like autonomous driving. As special cases of causal confusion, inertia problem (Codevilla et al., 2019) and copycat problem (Wen et al., 2020) are proposed to describe the strong reliance of an imitator's policy on the expert's previous actions, even when such actions are no explicitly provided as input.

Figure 1a illustrates the decision process of imitators suffering from inertia and copycat problems. $\boldsymbol{o}_{t-3:t}$ and $\boldsymbol{o}_{j:j+1}$ denote the observations recorded in successive frames, $a_{acc,i}$ is the expert's acceleration command in current frame i (i.e., $a_{acc,i} > 0$ means speed up, $a_{acc,i} < 0$ represents slow down), $v_{speed,i}$ is the speed in current frame. As discussed by (Codevilla et al., 2019), low speed often comes with negative acceleration in demonstrations, such a strong correlation tempts the inertia imitator to build a pattern: low speed causes braking. However, such a correlation exists only because the expert braked in previous frames. Moving a step forward, a copycat imitator seized that the variation in speed in previous frames provides clues of the expert's former action, and it turns to replicate the previous action to achieve low prediction error. However, this shortcut is also misleading, as in deployment time, the copycat imitator repeats its previous prediction.

Researchers (Cultrera et al., 2023; Guo et al., 2024; Seo et al., 2023; Wen et al., 2021; Samsami et al., 2021; Tien et al., 2022) have attributed such reliance on spurious correlations as (1) **The complexity of the task itself:** tasks like driving have sophisticated kinematics, diverse scenarios, continuous action space and numerous environmental parameters. (2) **The partial observability of state:** typical IL approaches to achieve driving (Hu et al., 2022a; Chuang et al., 2022) often design the expert to have access to the ground truth state (i.e., pre-trained RL agent with BEV observation or scripted expert with ground truth information), while the imitator can only receive the visual observation and measurement vector of ego vehicle. (3) **The lack of explicit causal model:** an imitator can not distinguish the causal and non-causal policies with similar offline evaluation performance, without

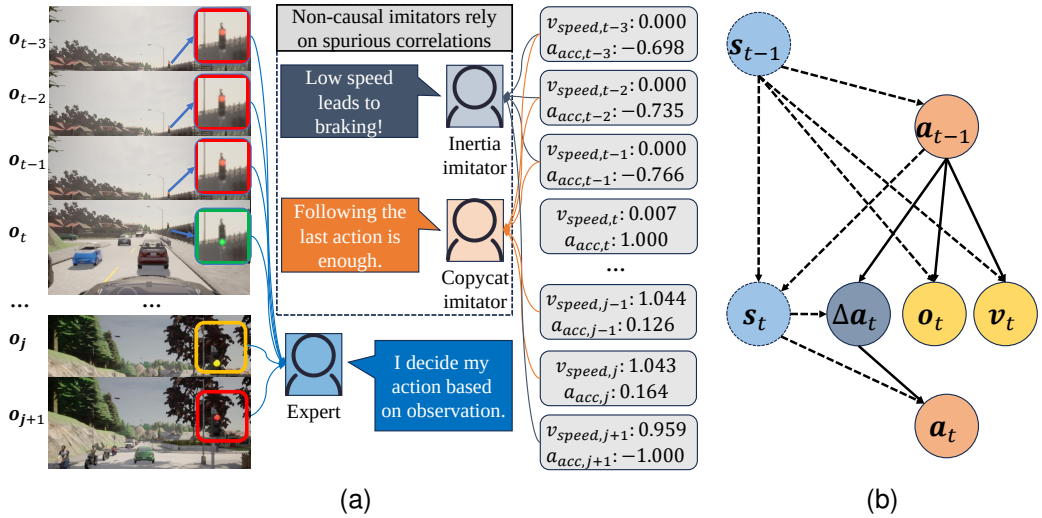

Figure 1: (a): An illustration of different decision-making patterns. In timestep t, a 3-tuple $(\boldsymbol{o}_t, \boldsymbol{v}_t, \boldsymbol{a}_t)$ is recorded as expert demonstration. $\boldsymbol{o}_t$ denotes the image captured by the front camera, $v_{speed,t}$ is measured speed of ego vehicle, and $a_{acc,t}$ is the expert's command in acceleration (< 0 means braking, > 0 means accelerate). (b): A causal diagram of the data-generating process within two timesteps $t-1, t$ in driving tasks, subscripts representing the timestep. Dashed nodes and edges denote the variables and interactions an imitator cannot observe: as modeled in Section 2.2, the mismatched observation forms impose difficulty in recovering expert policy. $\Delta \boldsymbol{a}_t$ is the difference between previous expert action $\boldsymbol{a}_{t-1}$ and current expert action $\boldsymbol{a}_t$. $\boldsymbol{v}_t$ denotes the measurement vector that comes with the image observation $\boldsymbol{o}_t$.

prior knowledge of the causal model of the task. (4) **The evident and consistent spurious correlations:** a mapping between spurious features and the expert's previous action is easily learned and rarely violated, while differences between successive actions are usually minor. With all the factors above, an imitator prefers to infer previous actions from spurious correlated features as its prediction, instead of struggling in the large network parameter search space.

Appendix A.1 provides an introduction to works in related areas. Although methods have proposed to bring causality into fields of IL and autonomous driving, most of their approaches either focus on theoretical analysis(Howard & Kunze, 2024; Ruan et al., 2022; Ruan & Di, 2022; Kumor et al., 2021; Swamy et al., 2022b), interpreting and evaluating agents' behavior (Maier et al., 2024; Li et al., 2024; Atakishiyev et al., 2023; Jacob et al., 2022; Hart & Knoll, 2020), operating in relatively simple settings (Guo et al., 2024; Samsami et al., 2021), or designed for certain sub-tasks (Cheng et al., 2024; Hu et al., 2022b; Tang et al., 2022) instead of end-to-end driving. In contrast, we aim to use causality to assist the imitator in visually complex end-to-end driving tasks.

2.2 Definitions

In this paper, we consider the imitation driving task under the partially observable Markov decision process (POMDP) setting. POMDP is commonly used to model decision-making problems in nondeterministic and partially observable scenarios.

Similar to (Kurniawati, 2022), the POMDP model is defined as a 5-tuple $< \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{T}, \mathcal{F} >$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ the action space, $\mathcal{O}$ the imitator's observation space, $\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ denotes the state transition function, and $\mathcal{F}(\boldsymbol{o}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ denotes the observation function. Here, variables' subscripts represent the timesteps they are in. We use boldface with lowercase letters to denote instances in corresponding spaces, such as $\boldsymbol{s}_t \sim \mathcal{S}$, $\sim$ denotes the sample process.

The partial observability in the driving task is represented by the mismatch between the expert's and the imitator's observation form. While the expert receives and processes the ground truth information (e.g. bird-eye view images or vectored description of the whole scenario) as state $\boldsymbol{s}$, the

imitator can only observe an image $\boldsymbol{o}$ as the profile of the current expert state, and $\boldsymbol{o}$ only carries partial information about $\boldsymbol{s}$.

Additionally, we consider the driving task under the conditional imitation learning (CIL) setting proposed by (Codevilla et al., 2018), and introduce a measurement vector $\boldsymbol{v}$ that come with $\boldsymbol{o}$. $\boldsymbol{v}$ describes the ego vehicle state and navigation information, and is also provided to the imitator as conditions. As human drivers often drive under the indication of navigation information, we also design our method to express the effect of $\boldsymbol{v}$ more comprehensively to the imitator. In the implementation, the navigation information can be derived from a simple route planner which requires no parameter optimization, and the incorporation of $\boldsymbol{v}$ in CIL does not violate the general end-to-end driving setting.

At each timestep t the expert observes the state $\boldsymbol{s}_t$, selects an action $\boldsymbol{a}_t \sim \pi_e(\boldsymbol{a}_t|\boldsymbol{s}_t)$ based on its policy π_e, and then observe the next state $\boldsymbol{s}_{t+1} \sim \mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ sampled from the state transition function. During the above interaction with environment, image observation $\boldsymbol{o}_t$ and measurements $\boldsymbol{v}_t$ that obtained from the observation function $(\boldsymbol{o}_t, \boldsymbol{v}_t) \sim \mathcal{F}(\boldsymbol{o}_t, \boldsymbol{v}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$ are recorded as the proxy of the state $\boldsymbol{s}_t$ observed by expert. The dataset $\mathbf{D_e}$ is organized as a combination of N expert demonstrations $(\boldsymbol{o}_i, \boldsymbol{v}_i, \boldsymbol{a}_i)_{i=1}^N$. The goal of typical BC approaches is to learn a policy $\pi(\boldsymbol{a}_t|\boldsymbol{o}_t, \boldsymbol{v}_t)$ with the supervision of the expert demonstrations such that the distribution of hidden reward generated by the policy π is the same as the one generated by expert policy π_e.

Following the setting of typical Behavioral Cloning from Observation Histories (BCOH) approaches (Chuang et al., 2022; Hu et al., 2022a; Seo et al., 2023), we assume that observations in history can provide useful information about $\boldsymbol{s}_t$, as a snapshot typically cannot tell the whole story. Therefore, we extend the temporal input range of π to encourage it to extract more information from the past and have a better understanding of the current scenario. By setting a history perception window length l, we use $\boldsymbol{o}_{t-l:t}$ to denote the observed images in the period $[t-l, t]$, $\boldsymbol{v}_{t-l:t}$ is defined in a similar way. In a certain timestep t, the observation history under the perception window length l of an imitator is defined as $\boldsymbol{h}_t = (\boldsymbol{o}_{t-l:t}, \boldsymbol{v}_{t-l:t})$, the imitator's policy is rewrote π as $\pi(\boldsymbol{a}_t|\boldsymbol{h}_t)$, and the training dataset is organized as $\mathbf{D_e} = (\boldsymbol{h}_i, \boldsymbol{a}_i)_{i=1}^N$.

While the incorporation of observations in the history provides vast information and helps imitators learn the dynamics of the environment, it also introduces shortcuts in imitating and prevents imitators from faithfully recovering π_e, as showcased by Figure 1a.

3 METHOD

3.1 INSIGHTS FROM CAUSALITY

We use the causal diagram to give a description of the driving tasks, as shown in Figure 1b. In the modeled causal graph, it's clear to see that the observed variable tuple $(\boldsymbol{o}_t, \boldsymbol{v}_t)$ has not directed edges that point at the current state $\boldsymbol{s}_t$ and action $\boldsymbol{a}_t$. We design this based on the fact that $(\boldsymbol{o}_t, \boldsymbol{v}_t)$ is just a profile of the state observed by the expert, derived as $(\boldsymbol{o}_t, \boldsymbol{v}_t) \sim \mathcal{F}(\boldsymbol{o}_t, \boldsymbol{v}_t|\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1})$. Therefore, directly building a policy that maps $(\boldsymbol{o}_t, \boldsymbol{v}_t)$ to $\boldsymbol{a}_t$ is inappropriate, as the state $\boldsymbol{s}_t$ that the expert used for the decision has not been inferred, and $(\boldsymbol{o}_t, \boldsymbol{v}_t)$ cannot cover enough information for replicating $\pi_e(\boldsymbol{a}_t|\boldsymbol{s}_t)$. On the other hand, building an imitator policy $\pi(\boldsymbol{a}_t|\boldsymbol{h}_t)$ that considers historical information can also foster the negative effects of previous actions on identifying the expert's decision process.

In this paper, we propose Causality-inspired Contrastive Conditional Imitation Learning (3CIL), an BC approach that incorporates causal insights into its design. In the training stage, 3CL decomposes the task of IL into two sub-tasks: representation learning and policy learning, corresponding to learning a representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$, and a predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$. Here, we add hats '$\hat{}$' to the imitator's predictions to distinguish them from the expert's states and actions. Based on the causal graph and analysis we made above, we conclude the important traits (**T1**,**T2**, **T3**) that a robust imitator must have, and introduce the targeted treatments in our proposed method 3CIL, as shown in following paragraphs.

(T1) Ability to extract enough information from observation history, bridging a mapping $\boldsymbol{h}_t \rightarrow \hat{\boldsymbol{s}}_t$ matches $(\boldsymbol{s}_{t-1}, \boldsymbol{a}_{t-1}) \rightarrow \boldsymbol{s}_t$: a robust imitator must have its clues about the current scenario. To achieve **T1**, 3CIL imposes a future image reconstruction task on its representation learning

phase. With emphasis on the temporal and navigation conditions, we propose to model the feature extraction module with a conditional Variational Auto-Encoder (VAE) and recurrent process. Based on supervision from the causal direction $\hat{\boldsymbol{s}}_t \rightarrow \hat{\boldsymbol{o}}_{t+1}$, 3CIL transforms the history concluding process into a simulation of observation function $\mathcal{F}(\boldsymbol{o}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ in the modeled POMDP. With this modification, the representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is urged to conclude enough information from observation history $\boldsymbol{h}_t$, so that it can match with the actual expert state $\boldsymbol{s}_t$ in the metric of quality of inferred future image observation $\hat{\boldsymbol{o}}_t$.

(T2) Minor reliance on spurious correlations, learning influence of previous actions through $\boldsymbol{a}_{t-1} \rightarrow \boldsymbol{s}_t \rightarrow \boldsymbol{a}_t$ instead of $\boldsymbol{h}_t \leftarrow \boldsymbol{a}_{t-1} \rightarrow \boldsymbol{a}_t$: to guarantee its performance in non-i.i.d deployment time. To accomplish **T2**, 3CIL incorporates an action residual prediction task in the representation learning phase to encourage the model to capture changes in the expert's decisions. The proposed action residual prediction task enforces the imitator to capture the variations within consecutive actions $\Delta\boldsymbol{a}_t = \boldsymbol{a}_t - \boldsymbol{a}_{t-1}$ without explicitly accessing $\boldsymbol{a}_t$ and $\boldsymbol{a}_{t-1}$, which bypasses spurious correlation by the effect estimation in causal direction $\hat{\boldsymbol{s}}_t \rightarrow \Delta\hat{\boldsymbol{a}}_t$ (i.e., require $\boldsymbol{s}_t$ to reflect clues about changes between actions, instead of serve as proxy of $\boldsymbol{a}_{t-1}$). We also add a contrastive learning objective to help shape a regression-aware representation space that provides clues about current action. The supervised contrastive learning guides the representation model with the anti-causal direction $\hat{\boldsymbol{s}}_t \leftarrow \boldsymbol{a}_t$ hindsight, to enhance the consistency of causal effect from $\hat{\boldsymbol{s}}_t$ to predicting action $\boldsymbol{a}_t$.

(T3) Ability to investigate the difference between the expert's policy and the imitator's policy, identifying scenarios that caused high divergence between $\boldsymbol{a}_t$ and $\pi(\boldsymbol{a}_t|\boldsymbol{h}_t)$: a robust imitator must not be satisfied with its great average offline evaluation performance, but to focus on the factors that caused its inconsistency with expert behaviors. For **T3**, 3CIL proposes a sample weight term to guide the imitator focusing on samples that cause their predictions to contradict the expert.

In conclusion, the above three traits indicate stable causal influences: $\boldsymbol{h}_t \rightarrow \hat{\boldsymbol{s}}_t$ and $\hat{\boldsymbol{s}}_t \rightarrow \hat{\boldsymbol{a}}_t$, to simulate the actual expert strategy. Figure 2 and Section 3.2 describe the treatments used in 3CIL that enhance the robustness of the representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ from both causal direction ($\hat{\boldsymbol{s}}_t \rightarrow \Delta\hat{\boldsymbol{a}}_t, \hat{\boldsymbol{s}}_t \rightarrow \hat{\boldsymbol{o}}_{t+1}$) and anti-causal direction ($\hat{\boldsymbol{s}}_t \leftarrow \boldsymbol{a}_t$). With supervision from these two directions, the extracted representation $\hat{\boldsymbol{s}}_t$ is impulsed to have steady causal effects on the descendant nodes of actual expert state $\boldsymbol{s}_t$, help aligning inferred state $\hat{\boldsymbol{s}}_t$ with $\boldsymbol{s}_t$ to produce a stable representation. Figure 3 and Section 3.3 illustrate the optimizing process of predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$. The incorporated sample-weighting term is inspired by classical studies in causal reasoning, to mitigate the biases in both the representation learning stage and expert demonstration distribution. We detail the design of learning objectives in 3CIL in the following sections.

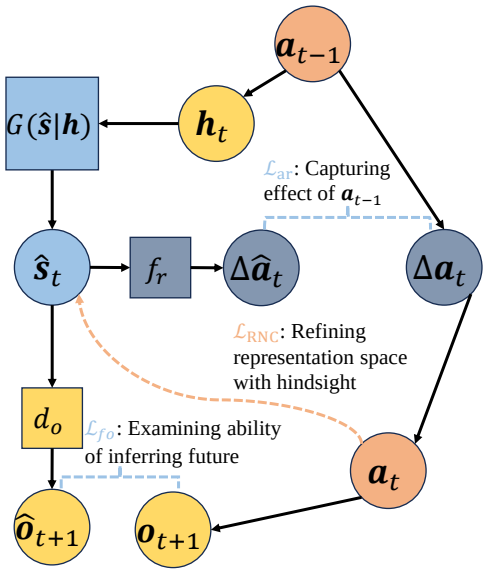

Figure 2: Modules in our 3CIL method are represented by rectangles, and colored based on the type of variables they are about to predict. 3CIL optimizes its representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ from both causal direction ($\mathcal{L}_{\text{ar}} : \hat{\boldsymbol{s}}_t \rightarrow \Delta\hat{\boldsymbol{a}}_t, \mathcal{L}_{fo} : \hat{\boldsymbol{s}}_t \rightarrow \hat{\boldsymbol{o}}_{t+1}$) and anti-causal direction $\mathcal{L}_{\text{RNC}} : \hat{\boldsymbol{s}}_t \leftarrow \boldsymbol{a}_t$. Here, f_r is the action residual predictor: $\Delta\hat{\boldsymbol{a}}_t = f_r(\hat{\boldsymbol{s}}_t)$, and d_o is an image decoder predicts the image in next frame :$\hat{\boldsymbol{o}}_{t+1} = d_o(\hat{\boldsymbol{s}}_t)$.

3.2 Representation Learning

The idea of representation learning is to train a representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ that extracts meaningful and reliable features for downstream predictor, corresponding to accomplish both **T1** and **T2**. Appendix A.2.1 provides an introduction to the implementation details and a visualization of modules in $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$.

As shown in (Codevilla et al., 2018), CIL eases the driving task by introducing conditions, i.e. navigation information organized as route commands that indicate what high-level action the agent

should take in the current route (e.g. lane following, turn left, turn right), into the input field of policy. Here, we move a step forward: instead of appending route commands into the feature vector for downstream predictor, 3CIL processes the route command as factors that affect the state transition.

Inspired by approaches in causal inference (Nie et al., 2021; Schwab et al., 2020) that impose the influence of treatment variables into networks' parameters for more accurate estimations, we seek to integrate the effect of navigation condition in the whole process of the representation model. By embedding the measurement vector $\boldsymbol{v}$ (including ego vehicle speed, and route commands) into a feed-forward network that decides the parameters of the hidden state's posterior distribution, we merge the influence of navigation information and current speed into the state abstraction process. Therefore, by using these navigation conditions as proxy variables of actual states observed by only experts, 3CIL guides the representation model to have a more reliable estimation of the current state.

Concretely, we design our $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ as a conditional VAE, with a recurrent state sequence module (RSSM) proposed by (Hafner et al., 2019), to simulate the state transition function $\mathcal{T}(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. We first extract the dense features $(\boldsymbol{x}_{t-l:t}, \boldsymbol{m}_{t-l:t})$ from non-structured historical images $\boldsymbol{o}_{t-l:t}$ and raw measurement vectors $\boldsymbol{v}_{t-l:t}$, with a pre-trained image encoder E_o and a measurement vector encoder E_v in $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$, produced as $(\boldsymbol{x}_{t-l:t}, \boldsymbol{m}_{t-l:t}) = [E_o(\boldsymbol{x}_{t-l:t}|\boldsymbol{o}_{t-l:t}), E_v(\boldsymbol{m}_{t-l:t}|\boldsymbol{v}_{t-l:t})]$. After that, we model the latent representation $\hat{\boldsymbol{s}}_t$ as conditioned on these features.

As empirically shown in (Hu et al., 2022a; Hafner et al., 2019), using both deterministic and stochastic features to model the latent representation enhances the flexibility and capability of the representation model. Therefore, we design the estimated latent state in timestep t: $\hat{\boldsymbol{s}}_t$ as a combination of the deterministic historical features $\boldsymbol{c}_t$ and the stochastic current latent information $\boldsymbol{z}_t$. RSSM models the distribution of current latent information $\boldsymbol{z}_t$ under transition $\mathcal{T}$ by a posterior distribution $q_z(\boldsymbol{z}_t|\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t) : \boldsymbol{z}_t \sim \mathcal{N}(\mu_\theta(\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t), \sigma_\theta(\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t))$ that conditioned on historical information $\boldsymbol{c}_t$, and features $(\boldsymbol{m}_t, \boldsymbol{x}_t)$ that derived from measurement vector and image observation. Here, the mean μ_θ and standard deviation σ_θ of modeled Gaussian distribution q_z are predicted by a feed-forward network that takes $(\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t)$ as input. The historical features $\boldsymbol{c}_t$ are extracted from a recurrent network f_d that takes former historical information $\boldsymbol{c}_{t-1}$ and former latent information $\boldsymbol{z}_{t-1}$ as input. For a training set $\mathbf{D_e}$ with N samples, the optimization objective is written in a variational lower bound form:

$$ELBO = \sum_{i=1}^{N-1} \underbrace{\mathbb{E}_{G(\boldsymbol{z}_i, \boldsymbol{c}_i|\boldsymbol{h}_i)}[logp(\boldsymbol{o}_{i+1}|\boldsymbol{z}_i, \boldsymbol{c}_i)]}_{\mathcal{L}_{fo}\text{: future image reconstruction}} - \underbrace{\mathrm{D}(q_z(\boldsymbol{z}_i|\boldsymbol{c}_i, \boldsymbol{m}_i, \boldsymbol{x}_i)||p_z(\boldsymbol{z}_i|\boldsymbol{c}_{i-1}, \boldsymbol{z}_{i-1}))}_{\text{posterior regularization}}, \quad (1)$$

where $\mathrm{D}(\cdot||\cdot)$ denotes the KL-divergence measurement, and the future image reconstruction task $\mathcal{L}_{fo}$ is carried out by an image decoder that receives $(\boldsymbol{z}, \boldsymbol{c})$ as input. Unlike previous works that reconstruct all images in $\boldsymbol{h}_t$ to examine the ability of $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ in reserving all history information, we resort to future image reconstruction $\mathcal{L}_{fo}$ to evaluate abilities of $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ to extract history information and infer future state. With this modification, $\mathcal{L}_{fo}$ indicates the representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ to capture temporal evolution by examining the simulated future image $\hat{\boldsymbol{o}}_{t+1}$. Therefore, $\hat{\boldsymbol{s}}$ is urged to reproduce the similar causal effect of $\boldsymbol{s}_t$ on $\boldsymbol{o}_{t+1}$.

Different from previous approaches (Hu et al., 2022a; Hafner et al., 2019), we eliminate the explicit use of the previous action variable $\boldsymbol{a}_t$ in all modules in the representation model, which seems to contradict the causal effect $\boldsymbol{a}_{t-1} \to \boldsymbol{s}_t$ modeled in the causal diagram Figure 1b. However, the causal diagram further shows that the effect of previous action $\boldsymbol{a}_{t-1}$ on current action $\boldsymbol{a}_t$ can be inferred from the variation between them, denotes as $\Delta\boldsymbol{a}_t = \boldsymbol{a}_t - \boldsymbol{a}_{t-1}$. In this view, we proposed to maximize the conditional mutual information $I(\hat{\boldsymbol{s}}_t, \boldsymbol{a}_t|\boldsymbol{a}_{t-1})$, by optimizing the prediction accuracy of $\Delta\boldsymbol{a}_t$ by $\hat{\boldsymbol{s}}$ only. Therefore, we introduce an action residual prediction objective in the representation learning phase:

$$\mathcal{L}_{\text{ar}} = \frac{1}{N}\sum_{i=1}^{N}(\Delta\boldsymbol{a}_i - f_r(\hat{\boldsymbol{s}}_i))^2, \quad (2)$$

where f_r is an action residual predictor that predicts action difference $\Delta\boldsymbol{a}_t = \boldsymbol{a}_t - \boldsymbol{a}_{t-1}$. The introduced action residual prediction task builds a causal directed edge $\hat{\boldsymbol{s}}_t \to \Delta\hat{\boldsymbol{a}}_t$ to encourage $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ capturing variations caused by previous actions.

While embedding the influence of observation history and action residual into the representation learning enhances the model's ability to infer the current state, it inevitably introduces more prominent spurious correlations, as discussed in former sections. Therefore, to alleviate such effects, we further introduce a contrastive learning objective to help shape a robust representation model.

As we aim to achieve driving with expert demonstrations, we resort to the supervised contrastive learning (Khosla et al., 2020) methods to enhance robustness, as the expert's actions naturally reflect differences among samples. With this intuition, we introduce the Rank-N-Contrast (RNC) loss from (Zha et al., 2023) to the optimization objective of representation learning in 3CIL. The idea of RNC is to align distances in the representation space ordered by distances in their labels, which by design meets the requirement of alleviating the effect of spurious correlation left in the features: samples that carry similar historical information may end up poles apart in the representation space when their corresponding action labels are different.

Therefore, $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ equipped with the RNC loss is encouraged to use the anti-causal relation $\hat{\boldsymbol{s}}_t \leftarrow \boldsymbol{a}_t$ to infer a suitable state corresponding to the actual expert action $\boldsymbol{a}_t$. In this view, the performed supervised contrastive learning is equivalent to conducting a hindsight investigation on the consistency of constructed causal effect $\hat{\boldsymbol{s}}_t \rightarrow \boldsymbol{a}_t$.

For a batch sampled from $\mathbf{D_e}$ with batch size B, we apply two independent augmentations to obtain a new batch with $2B$ samples, and write the RNC loss as:

$$\mathcal{L}_{\text{RNC}} = \frac{1}{2B}\sum_{i=1}^{2B}\frac{1}{2B-1}\sum_{j=1,j\neq i}^{2B} -\log\frac{\exp(\text{sim}(\hat{\boldsymbol{s}}_i,\hat{\boldsymbol{s}}_j)/\tau)}{\sum_{\hat{\boldsymbol{s}}_k\in set(i,j)}\exp(\text{sim}(\hat{\boldsymbol{s}}_i,\hat{\boldsymbol{s}}_k)/\tau)}\,, \tag{3}$$

where τ is the temperature parameter, $\text{sim}(\cdot)$ is the similarity measure between two inferred states (cosine similarity is used in this work), $set(i,j)$ collect those samples' representations whose corresponding action labels have higher rank (in terms of distance with $\hat{\boldsymbol{s}}_i$) compare to $\hat{\boldsymbol{s}}_j$ (i.e., $set(i,j) = \hat{\boldsymbol{s}}_k | k \neq i, \text{d}(\boldsymbol{a}_k,\boldsymbol{a}_i) \geq \text{d}(\boldsymbol{a}_j,\boldsymbol{a}_i)$, $\text{d}(\cdot,\cdot)$ measures distance between two labels).

After introducing all designs we proposed that aim to shape a robust representation model which meets the requirements **T1** and **T2**, we write our final optimization objective of $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ as maximizing: $\mathcal{L}_G = \mathit{ELBO} - \mathcal{L}_{\text{ar}} - \mathcal{L}_{\text{RNC}}$.

3.3 Policy Learning

After representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is trained and its parameters are frozen, the downstream predictor network $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ is more resistant to spurious correlations. However, a severe problem remains unsolved: the inconsistency of imitators when replicating an expert's strategy in certain scenarios. We believe this problem can be attributed to the unmatched growth between the diversity of driving scenarios and the number of expert demonstrations: scenarios are not distributed uniformly in the dataset. Such a characteristic indulges the imitator to be indifferent about rare scenarios and hinders it from achieving **T3**.

To enhance the imitator's ability to cope with rare situations, 3CIL incorporates a sample-weighting process in training the predictor $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$: i.e., we assign weights on samples based on the divergence between imitator and expert, enforcing the predictor attaching importance on rare situations. To detect the extent of divergence, we reuse the action residual predictor f_r that was trained from the previous phase, to describe such difference in a per-sample manner.

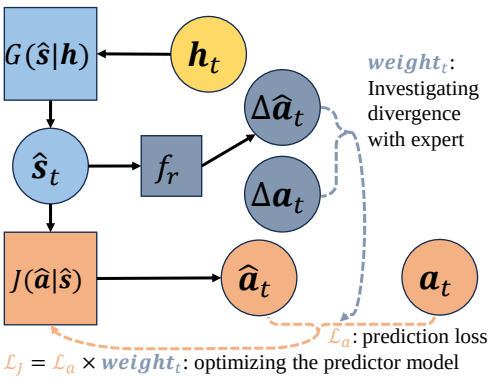

Figure 3: Modules in our 3CIL method are represented by rectangles, and colored based on the type of variables they are about to predict. 3CIL incorporates the sample-weighting term $\boldsymbol{weight}_t$ into the optimization objective of predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$, to transform the divergence between the expert and imitator into emphasis on important samples.

Denote the action residual prediction error between $f_r(\hat{\boldsymbol{s}})$ and $\Delta\boldsymbol{a}$ as $\delta\boldsymbol{a}$: $\delta\boldsymbol{a} = |\Delta\boldsymbol{a} - f_r(\hat{\boldsymbol{s}})|$, and denote the operation: bound$(\cdot, b_{min}, b_{max})$ as a function bounds a variable or all elements within a vector, to the range $[b_{min}, b_{max}]$. For a sample $(\boldsymbol{h}_i, \boldsymbol{a}_i)$ from expert demonstration, its corresponding weight is computed as:

$$\boldsymbol{weight}_i = \exp(\text{bound}(\delta\boldsymbol{a}_t - \overline{\delta\boldsymbol{a}}, b_{min}, b_{max}) \times \gamma), \tag{4}$$

where $\overline{\delta\boldsymbol{a}}$ is the mean of residual errors in the minibatch, $[b_{min}, b_{max}]$ is set to $[-0.3, 0.3]$, γ is the factor controlling strength of sample weight and it is set to 6.67 in our experiment.

Consider the causal relations between $(\boldsymbol{s}_t, \Delta\boldsymbol{a}_t, \boldsymbol{a}_t)$ in Figure 1b, variations in $\delta\boldsymbol{a}_t$ can be seen as performing interventions on $\Delta\boldsymbol{a}_t$, such variation based sample-weighting process is akin to techniques like inverse probability weighting and doubly robust learning that widely used in the causal inference literature. Indeed, the quantification of prediction error for $\Delta\boldsymbol{a}_t$ identifies the potentially under-represented certain scenarios and also mediates the bias introduced by inaccurate estimations in previous representation learning.

The training objective of predictor $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ is then to minimize: $\mathcal{L}_J = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_a(\boldsymbol{a}_t, J(\hat{\boldsymbol{s}}_i)) \times \boldsymbol{weight}_i$, where $\mathcal{L}_a$ can be typical metrics that used for supervised learning (e.g., cross-entropy and mean-squared error, based on the form of expert action label).

3CIL divides the imitating process as two separate stages: first use $\mathcal{L}_G$ to train a representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$, then $\mathcal{L}_J$ is used to train a predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ while the parameters of $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is frozen. We detail the implementation of 3CIL method in Appendix A.2.For a certain sample $\boldsymbol{h}_i$, the imitator's prediction is made as: $\hat{\boldsymbol{a}}_i \sim J(\hat{\boldsymbol{a}}_i|\hat{\boldsymbol{s}}_i), \hat{\boldsymbol{s}}_i \sim G(\hat{\boldsymbol{s}}_i|\boldsymbol{h}_i)$.

4 Experimental Evaluation

4.1 Settings

Environments and dataset: We conduct in the visually complex driving simulator CARLA (Dosovitskiy et al., 2017). Expert demonstrations are collected by an RL agent from (Zhang et al., 2021b) that trained using privileged information as input, we deploy it in four towns (Town01, Town03, Town04, Town06) to generate about $N = 1,125,300$ training samples. These samples is then organized in form of $\mathbf{D_e} : (\boldsymbol{h}_i, \boldsymbol{a}_i)_{i=1}^{N}$, with the perception window length l set to 4, and the observation history $\boldsymbol{h}_t$ is organized as $\boldsymbol{h}_t = (\boldsymbol{o}_{t-l:t}, \boldsymbol{v}_{t-l:t})$. An image observation $\boldsymbol{o}$ is of size $(\text{channel=RGB}, \text{width=240px}, \text{height=150px})$, and a measurement vector $\boldsymbol{v}$ is composed as $\boldsymbol{v} = (v_{speed}, v_{route\_command}, v_{route\_command\_next})$.

In the testing phase, we modify the weather conditions, traffic density, camera parameters, and also introduce two new towns (Town02, Town05) as environments to evaluate the performance of methods under severe distribution shifts. We design four evaluation scenario settings (denote as Scenario 1,2,3,4) corresponding to experiments in four towns that are seen in $\mathbf{D_e}$, and two evaluation scenario settings (denote as Scenario 5,6) for experiments in Town02 and Town05. Appendix A.3.3 and A.3.4 described the experiment design in detail.

Baselines: For comparison, we choose the following baselines as representatives of different types of approaches. (1) Conditional Imitation Learning (**CIL**, (Codevilla et al., 2018)): stands as the vanilla CIL method. (2) Keyframe-Focused Visual Imitation Learning (**Keyframe**, (Wen et al., 2021)): utilizes action prediction errors to assign weights on samples. (3) Domain Generalizable Imitation Learning by Causal Discovery (**DIGIC**, (Chen et al., 2024)): performs causal discovery to sort causal features and learn a domain generalizable policy. (4) Past Action Leakage Regularization (**PALR**, (Seo et al., 2023)): imposes regularization on conditional dependence between inferred state $\hat{\boldsymbol{s}}_t$ and previous action $\boldsymbol{a}_{t-1}$ to alleviate spurious correlation. (5) **Premier-TACO** from (Zheng et al., 2024): conducts temporal action-driven contrastive learning to shape a robust representation model. We also implement two methods that can be seen as ablation experiments of 3CIL. (6) Rank-N-Contrast framework (**RNC**, (Zha et al., 2023)): adding the $\mathcal{L}_{\text{RNC}}$ loss into the training representation model, without action residual prediction task. (7) Visual Imitation Learning via Residual Action Prediction (**RAP**, (Chuang et al., 2022)): adding the $\mathcal{L}_{\text{ar}}$ in training representation, without assigning

sample weights in training predictor.A detailed introduction about used baselines and the ways we implement them is provided in Appendix A.3.1.

Evaluation metrics: We quantify the performance of methods based on three metrics: accumulated rewards, average collision rate, and average speed. The reward function is constructed as $R = r_{speed} + r_{position} + r_{rotation} + r_{action}$, a combination of four factors that independently judge an agent's driving ability in following indicated routes, similar to previous work (Zhang et al., 2021b). Details of the reward function are listed in Appendix A.3.5. We combine all metrics to discuss strategies learned by each method.

4.2 Performance and Discussion

Table 1: Performance of each method in three metrics: accumulated reward (R), average collision rate (C, in ‰), and average speed (S, in km/h). We add arrows beside R and C to indicate the optimal direction. No arrow is placed beside S, as the speed metric alone can not reflect driving performance. **Bold** numbers in rows R and C indicate the best results, second-best results are underlined.

Metric \ Method		CIL	Keyframe	DIGIC	PALR	Premier-TACO	RNC	RAP	3CIL(Ours)
Scenario 1	R ↑	330.49	353.83	437.32	354.47	469.99	411.31	383.54	**521.26**
	C ↓	0.66	0.58	0.73	2.42	0.85	0.55	0.60	**0.54**
	S	5.22	6.05	11.99	8.52	12.59	7.50	7.95	9.76
Scenario 2	R ↑	12.14	309.70	484.49	422.79	431.22	519.14	362.31	**587.44**
	C ↓	**0.36**	0.57	0.47	1.67	0.55	0.42	0.53	0.46
	S	7.89	9.56	18.96	19.11	19.76	18.12	15.95	19.85
Scenario 3	R ↑	247.29	125.30	404.44	327.85	204.38	64.80	136.6	**420.38**
	C ↓	1.35	1.56	1.38	4.18	1.31	2.20	3.15	**1.25**
	S	3.99	7.74	13.43	9.60	10.76	12.67	11.66	12.07
Scenario 4	R ↑	345.00	529.68	400.42	837.98	561.13	735.19	505.63	**966.35**
	C ↓	0.37	0.31	0.32	0.97	0.37	0.31	0.35	**0.27**
	S	7.00	9.46	18.58	11.23	19.40	17.05	16.78	16.20
Scenario 5	R ↑	7.18	278.95	306.10	421.16	516.70	299.72	302.50	**538.50**
	C ↓	**0.29**	0.53	0.49	1.47	0.37	0.50	0.59	0.48
	S	7.61	9.24	12.60	11.92	15.32	13.71	11.00	14.46
Scenario 6	R ↑	45.93	215.77	409.88	389.07	331.29	**447.44**	195.53	447.24
	C ↓	**0.34**	0.64	0.94	1.54	0.68	0.64	0.63	0.59
	S	4.32	8.95	10.98	8.66	12.19	8.56	7.78	10.99

In Table 1, we present the evaluation results in the CARLA simulator. A detailed ablation study is provided in Appendix A.4. We analyze the performance of methods from several aspects.

Effect of spurious correlation. Our proposed method 3CIL is one of the most cautious drivers with the lowest collision rate in half settings (3 of 6). Interestingly, another method with the lowest collision rate is the earliest approach CIL Codevilla et al. (2018) which did not consider the spurious correlations problem. This phenomenon can be explained when we consider the accumulated reward and average speed: the quantized results of CIL in these two metrics are significantly lower than most of its competitors, suggesting that CIL has built doubtful decision patterns that showed overly cautiousness. Indeed, in our observation, the agent trained with CIL often got stuck or even failed to launch. In contrast, the rest of the methods generally have no such issues, demonstrating the necessity of alleviating the effects of spurious correlations.

On the contrary, PALR effectively removes the effect from the previous action by regularizing conditional dependence $(\hat{\boldsymbol{s}}_t; \boldsymbol{a}_{t-1} | \boldsymbol{a}_t)$. However, such regularization may not be suitable for driving tasks, as shown in Figure 1b, investigation of effects from previous action is still required for recovering information about the expert's state. Indeed, although PALR achieves relatively stable rewards, it obtains the highest collision rates in most settings.

Sample-weighting strategies and contrastive learning help imitating. Both Keyframe and 3CIL translate errors in prediction into assigned weights on corresponding samples. Experimental results demonstrate that sample-weighting strategies indicate the imitator to focus on crucial changepoints

in a simple yet efficient way. Moreover, the proposed weighting design in 3CIL utilizes errors in action residual prediction in the representation learning stage instead of the copycat policy action prediction errors in (Wen et al., 2021). Such a strategy enables 3CIL to identify abnormal scenes that the representation model fails to cover and empowers it to cope with more general problems than the copycat phenomenon.

On the other hand, contrastive learning also contributes the imitating performance. Both Premier-TACO (temporal action-driven contrastive loss) (Zheng et al., 2024) and RNC (supervised contrastive loss) (Zha et al., 2023) can assist the representation model to capture the intrinsic characteristics from observation history, which is demonstrated by their relatively good performance in both seen and unseen scenarios. Our 3CIL approach utilizes supervised contrastive learning to help infer states, as the actions made by experts can provide clear clues for constructing representation space, without the need to tune hyper-parameters for sampling temporal positive/negative pairs.

While methods like DIGIC that introduce causal discovery can also investigate the essential relationships related to experts' decisions, the masking or filtering operation will inevitably cause information loss, which may lead to safety concerns in applications like driving, and sub-optimal performance in evaluation.

Robustness requires effort. Although variations in experimental settings and novel map layouts pose challenges to imitators' capabilities, 3CIL still maintains a robust driving strategy. As shown in Table 1, 3CIL obtains the highest accumulated rewards in most settings (5 of 6). Recall that the reward function in evaluation measures the ability of an imitator in executing general driving tasks given navigation conditions, the outstanding performance in accumulated rewards indicates that the pursuit for **T1,T2,T3** does improve the robustness of the imitator.

Moreover, when we analyze the performance of RNC and RAP that can be seen as parts of the ablation study of 3CIL, the effectiveness of interactions we introduced in Figure 2 and Figure 3 become evident: the shared reconstruction task $\mathcal{L}_{fo}$, coupled with hindsight from supervised contrastive learning task $\mathcal{L}_{\text{RNC}}$ or action variation capturing task $\mathcal{L}_{\text{ar}}$, both alleviate the spurious correlations, but fail to maintain steady performance in all settings. Concretely, when we incorporate the anti-causal direction $\hat{\boldsymbol{s}} \leftarrow \boldsymbol{a}_t$ influence to shape the imitator's representation space, $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is enforced arrange inferred states to match their corresponding potential actions' propensity, but not enforced to capture the effect that previous actions imposed on the current state, while RAP approach is equivalent to do the opposite. Either way, the absence of essential information will result in an inadequate estimation of the expert state, hindering models from robust generalization performance. Also, the sample-weighting process proposed in 3CIL does improve the imitator's ability to handle rare scenarios, as the agent trained with 3CIL shows more caution and obtains lower collision rates.

5 Conclusion

In this work, we investigate the factors hindering imitation learning methods from generalizing training performance into unfamiliar testing environments in autonomous driving tasks. Based on causal reasoning about the expert's decision process, we identify crucial traits an imitator must have for robust performance. After that, we introduce Causality-inspired Contrastive Conditional Imitation Learning (3CIL), an imitation learning method that imposes regularization on the imitator's representation by supervised contrastive learning and action residual prediction, corresponding to assigning supervisions on representation model from both causal direction and anti-causal direction to guarantee quality of the inferred state. Moreover, 3CIL introduces a sample-weighting term to transform the high divergences between the expert and imitator, into the emphasis on rare scenarios, enabling the imitator to adapt to diverse situations. We perform experiments in the CARLA simulator to demonstrate the effectiveness of the proposed 3CIL.

References

Shahin Atakishiyev, Mohammad Salameh, Housam Babiker, and Randy Goebel. Explaining autonomous driving actions with visual question answering. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1207–1214. IEEE, 2023.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018.

Yang Chen, Yitao Liang, and Zhouchen Lin. Digic: Domain generalizable imitation learning by causal discovery. *arXiv preprint arXiv:2402.18910*, 2024.

Debo Cheng, Jiuyong Li, Lin Liu, Kui Yu, Thuc Duy Le, and Jixue Liu. Toward unique and unbiased causal effect estimation from data with hidden variables. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving. *arXiv preprint arXiv:2404.14327*, 2024.

Pranav Singh Chib and Pravendra Singh. Recent advancements in end-to-end autonomous driving using deep learning: A survey. *IEEE Transactions on Intelligent Vehicles*, 2023.

Chia-Chi Chuang, Donglin Yang, Chuan Wen, and Yang Gao. Resolving copycat problems in visual imitation learning via residual action prediction. In *European Conference on Computer Vision*, pp. 392–409. Springer, 2022.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9329–9338, 2019.

Luca Cultrera, Federico Becattini, Lorenzo Seidenari, Pietro Pala, and Alberto Del Bimbo. Addressing limitations of state-aware imitation learning for autonomous driving. *arXiv preprint arXiv:2310.20650*, 2023.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pp. 11698–11709, 2019.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. In *Conference on Robot Learning*, pp. 1–16. PMLR, 2017.

Wiebke Günther, Urmi Ninad, and Jakob Runge. Causal discovery for time series from multiple datasets with latent contexts. *arXiv preprint arXiv:2306.12896*, 2023.

Jiayu Guo, Mingyue Feng, Pengfei Zhu, Jinsheng Dou, Di Feng, Chengjun Li, Ru Wan, and Jian Pu. Mitigating causal confusion in vector-based behavior cloning for safer autonomous planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 16475–16481. IEEE, 2024.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pp. 2555–2565. PMLR, 2019.

Patrick Hart and Alois Knoll. Counterfactual policy evaluation for decision-making in autonomous driving. *arXiv preprint arXiv:2003.11919*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Rhys Howard and Lars Kunze. Extending structural causal models for use in autonomous embodied systems. *arXiv preprint arXiv:2406.01384*, 2024.

Anthony Hu, Gianluca Corrado, Nicolas Griffiths, Zak Murez, Corina Gurau, Hudson Yeo, Alex Kendall, Roberto Cipolla, and Jamie Shotton. Model-based imitation learning for urban driving. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, pp. 20703–20716, 2022a.

Yeping Hu, Xiaogang Jia, Masayoshi Tomizuka, and Wei Zhan. Causal-based time series domain generalization for vehicle intention prediction. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7806–7813. IEEE, 2022b.

Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *The Journal of Machine Learning Research*, 21(1):3482–3534, 2020.

Paul Jacob, Éloi Zablocki, Hedi Ben-Younes, Mickaël Chen, Patrick Pérez, and Matthieu Cord. Steex: steering counterfactual explanations with semantics. In *European Conference on Computer Vision*, pp. 387–403. Springer, 2022.

Diviyan Kalainathan, Olivier Goudet, and Ritik Dutta. Causal discovery toolbox: Uncovering causal relationships in python. *Journal of Machine Learning Research*, 21(37):1–5, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 18661–18673, 2020.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Abhinav Kumar, Amit Deshpande, and Amit Sharma. Causal effect regularization: automated detection and removal of spurious correlations. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 20942–20984, 2023.

Daniel Kumor, Junzhe Zhang, and Elias Bareinboim. Sequential causal imitation learning with unobserved confounders. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 14669–14680, 2021.

Hanna Kurniawati. Partially observable markov decision processes and robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):253–277, 2022.

Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14128–14147, 2022.

Jiankun Li, Hao Li, Jiangjiang Liu, Zhikang Zou, Xiaoqing Ye, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. Exploring the causality of end-to-end autonomous driving. *arXiv preprint arXiv:2407.06546*, 2024.

Robert Maier, Lisa Grabinger, David Urlhart, and Jürgen Mottok. Causal models to support scenario-based testing of adas. *IEEE Transactions on Intelligent Transportation Systems*, 25(2): 1815–1831, 2024.

Lizhen Nie, Mao Ye, Qiang Liu, and Dan Nicolae. Vcnet and functional targeted regularization for learning causal effects of continuous treatments. *arXiv preprint arXiv:2103.07861*, 2021.

Pedro A Ortega, Markus Kunesch, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Joel Veness, Jonas Buchli, Jonas Degrave, Bilal Piot, Julien Perolat, et al. Shaking the foundations: delusions in sequence models for interaction and control. *arXiv preprint arXiv:2110.10819*, 2021.

Jongjin Park, Younggyo Seo, Chang Liu, Li Zhao, Tao Qin, Jinwoo Shin, and Tie-Yan Liu. Object-aware regularization for addressing causal confusion in imitation learning. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 3029–3042, 2021.

Junhyung Park and Krikamol Muandet. A measure-theoretic approach to kernel conditional mean embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 21247–21259, 2020.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Samuel Pfrommer, Yatong Bai, Hyunin Lee, and Somayeh Sojoudi. Initial state interventions for deconfounded imitation learning. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 2312–2319. IEEE, 2023.

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

Kangrui Ruan, Junzhe Zhang, Xuan Di, and Elias Bareinboim. Causal imitation learning via inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Mohammad Reza Samsami, Mohammadhossein Bahari, Saber Salehkaleybar, and Alexandre Alahi. Causal imitative model for autonomous driving. *arXiv preprint arXiv:2112.03908*, 2021.

Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning counterfactual representations for estimating individual dose-response curves. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5612–5619, 2020.

Seokin Seo, Hyeong Joo Hwang, Hongseok Yang, and Kee-Eung Kim. Regularized behavior cloning for blocking the leakage of past action information. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 2128–2153, 2023.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, James Weimer, and Insup Lee. Memory-consistent neural networks for imitation learning. *arXiv preprint arXiv:2310.06171*, 2023.

Xinwei Sun, Botong Wu, Xiangyu Zheng, Chang Liu, Wei Chen, Tao Qin, and Tie-Yan Liu. Recovering latent causal factor for generalization to distributional shifts. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 16846–16859, 2021.

Gokul Swamy, Sanjiban Choudhury, Drew Bagnell, and Steven Wu. Causal imitation learning under temporally correlated noise. In *International Conference on Machine Learning*, pp. 20877–20890. PMLR, 2022a.

Gokul Swamy, Sanjiban Choudhury, J. Bagnell, and Steven Z. Wu. Sequence model imitation learning with unobserved contexts. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17665–17676. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/708e58b0b99e3e62d42022b4564bad7a-Paper-Conference.pdf.

Chen Tang, Wei Zhan, and Masayoshi Tomizuka. Interventional behavior prediction: Avoiding overly confident anticipation in interactive prediction. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11409–11415. IEEE, 2022.

Jeremy Tien, Jerry Zhi-Yang He, Zackory Erickson, Anca D Dragan, and Daniel S Brown. Causal confusion and reward misidentification in preference-based reward learning. *arXiv preprint arXiv:2204.06601*, 2022.

Shenghua Wan, Yucen Wang, Minghao Shao, Ruying Chen, and De-Chuan Zhan. Semail: eliminating distractors in visual imitation via separated models. In *International Conference on Machine Learning*, pp. 35426–35443. PMLR, 2023.

Lingxiao Wang, Zhuoran Yang, and Zhaoran Wang. Provably efficient causal reinforcement learning with confounded observational data. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 21164–21175, 2021.

Tian-Zuo Wang and Zhi-Hua Zhou. Actively identifying causal effects with latent variables given only response variable observable. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 15007–15018, 2021.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 2564–2575, 2020.

Chuan Wen, Jierui Lin, Jianing Qian, Yang Gao, and Dinesh Jayaraman. Keyframe-focused visual imitation learning. In *International Conference on Machine Learning*, pp. 11123–11133. PMLR, 2021.

Chuan Wen, Jianing Qian, Jierui Lin, Jiaye Teng, Dinesh Jayaraman, and Yang Gao. Fighting fire with fire: avoiding dnn shortcuts through priming. In *International Conference on Machine Learning*, pp. 23723–23750. PMLR, 2022.

Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Disentangled representation learning via neural structural causal models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9593–9602, 2021.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: learning continuous representations for regression. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 17882–17903, 2023.

Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end simulated driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Jiaqi Zhang, Chandler Squires, and Caroline Uhler. Matching a desired causal state via shift interventions. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, pp. 19923–19934, 2021a.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 15222–15232, 2021b.

Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé Iii, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-TACO is a few-shot policy learner: Pretraining multitask representation via temporal action-driven contrastive loss. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61413–61431. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/zheng24g.html`.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

Wenxuan Zhu, Chao Yu, and Qiang Zhang. Causal deep reinforcement learning using observational data. *arXiv preprint arXiv:2211.15355*, 2022.

A Appendix

A.1 Related work

A.1.1 Imitation Learning for Autonomous Driving

Imitation Learning (IL) is widely used in autonomous driving (Le Mero et al., 2022; Bansal et al., 2018), as it requires few or zero actual interactions with the target environment. Classical literatures have divided IL into behavior cloning (BC) and inverse reinforcement learning (IRL). The idea of adversarial learning also introduces adversarial imitation learning (AIL) to the IL family. End-to-end autonomous driving approaches typically use BC in training, as BC does not need actual interactions with the environment, but seeks to learn driving patterns from numerous offline demonstrations(Chib & Singh, 2023).

However, the performance of BC is often problematic, which arises from the contradiction between the i.i.d assumption made by BC and the out-of-distribution (OOD) nature of driving tasks (Sridhar et al., 2023). Such a conflict leads to the compounding error that a BC imitator behaves unreliably when observing unfamiliar scenarios. Moreover, BC often suffers from causal confusion (de Haan et al., 2019), as its lack of the explicit causal model makes the imitator cannot tell the difference between spurious correlations and causal relations. This phenomenon becomes more severe when the BC imitator interacts with the environment in a sequential manner. Commonly the learned shortcuts (Wen et al., 2022) fail to apply in the test stage, or the BC imitator is even stuck in delusions caused by itself(Ortega et al., 2021).

A.1.2 REMEDIES FOR CAUSAL CONFUSION

Previous approaches have proposed several remedies for handling the phenomenon of causal confusion to obtain a robust imitator, including: randomly masking encoded discrete features (Park et al., 2021), incorporating additional supervisions or regularization on the encoder or predictor (Hu et al., 2022a; Kumar et al., 2023; Seo et al., 2023), querying experts about certain scenarios (de Haan et al., 2019) and performing interventions on the environment state or policy input (Pfrommer et al., 2023; Ruan & Di, 2022), filtering input features based on causal discovery (Chen et al., 2024; Samsami et al., 2021), maximizing certain bounds or mutual information to achieve deconfounding (Swamy et al., 2022a; Wan et al., 2023).

However, masking and regularizing encourage the imitator to be indifferent about spurious correlated features but may cause the loss of useful information. Adding extra supervision during training requires modifying the data-collecting process, and querying experts or intervening in the environment seems unfeasible in the driving task. While approaches with causal discovery match humans' instinct, they typically only work with low-dimensional data form (i.e., vectored observation), and the choice of causal discovery algorithm impacts the imitator's performance. Approaches aimed to deconfound may need more clues about the expert's policy and the dynamic of target environments or mainly contribute to the theoretical analysis. Therefore, we aim to develop a method that requires no extra information beyond the training dataset and has a robust policy that can drive in unseen environments without reliance on spurious correlations.

A.1.3 CAUSAL REASONING

Causal reasoning (Pearl, 2009) approaches can be generally divided into two genres: causal discovery and causal inference. Causal discovery aims to recover the underlying causal relations among variables in the target system, to help researchers learn the mechanisms of a system and aid downstream tasks. On the other side, causal inference is designed to learn the effect of modifying one/multiple variables' value (i.e., intervening on treatment variables) on the outcome variables, while considering the mechanisms between variables.

While typical studies have demonstrated the effectiveness and necessity of causality, incorporating causal reasoning into visually complex and partially observable tasks is still challenging. While efforts have been made to investigate causality in high-dimensional and confounded data (Günther et al., 2023; Zhu et al., 2022; Wang & Zhou, 2021; Wang et al., 2021; Sun et al., 2021; Cheng et al., 2022; Yang et al., 2021), the full identification of causalities in tasks like autonomous driving is still intractable without further assumptions (Zheng et al., 2018) or specifications (Huang et al., 2020). Moreover, using the observational samples alone generally cannot provide sufficient indications for recovering all causal relations or estimating precise causal influence. Performing interventions (Pfrommer et al., 2023; Zhang et al., 2021a) or querying experts about certain scenarios (de Haan et al., 2019; Zhang & Cho, 2017) are unfeasible in tasks with safety concerns and high interaction frequency like robotics and autonomous driving.

A.2 IMPLEMENTATION OF 3CIL

A.2.1 MODULES

As shown in Figure 4, $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is composed of an image encoder $E_o(\boldsymbol{x}|\boldsymbol{o})$, a measurement vector encoder $E_v(\boldsymbol{m}|\boldsymbol{v})$, a recurrent state sequence module (RSSM) from (Hafner et al., 2019) that combines both deterministic state model $f_d(\boldsymbol{c}_t|\boldsymbol{c}_{t-1}, \boldsymbol{z}_{t-1})$ and stochastic state model $q_z(\boldsymbol{z}_t|\boldsymbol{c}_t, \boldsymbol{x}_t, \boldsymbol{m}_t)$. $\boldsymbol{x}_t$ and $\boldsymbol{m}_t$ are the features extracted from the image encoder E_o and the measurement vector encoder E_v, respectively. $\boldsymbol{c}_t$ is a feature vector that preserves history information, and $\boldsymbol{z}_t$ is the hidden state sampled from a Gaussian distribution $p_s(\boldsymbol{z}_t|\boldsymbol{c}_t, \boldsymbol{x}_t, \boldsymbol{m}_t)$ whose mean and variance are parameterized by a feed-forward network.

Denote $\oplus$ as the concatenation operation, for a sample in timestep t with observation history $\boldsymbol{h}_t$, its corresponding representation is obtained as: $\hat{\boldsymbol{s}}_t = \boldsymbol{c}_t \oplus \boldsymbol{z}_t$, $\boldsymbol{c}_t \sim f_d(\boldsymbol{c}_t|\boldsymbol{c}_{t-1}, \boldsymbol{z}_{t-1}), \boldsymbol{z}_t \sim q_z(\boldsymbol{z}_t|\boldsymbol{c}_t, \boldsymbol{x}_t, \boldsymbol{m}_t)$, and we assume a fixed initial feature vector $\boldsymbol{c}_0$ without loss of generality.

In addition, we incorporate an image decoder $d_o(\hat{\boldsymbol{o}}_{t+1}|\hat{\boldsymbol{s}}_t)$ to carry out the image reconstruction task, and an action residual predictor $f_r(\hat{\Delta}\boldsymbol{a}_t|\hat{\boldsymbol{s}}_t)$ to capture the variation in expert's actions in a period. In

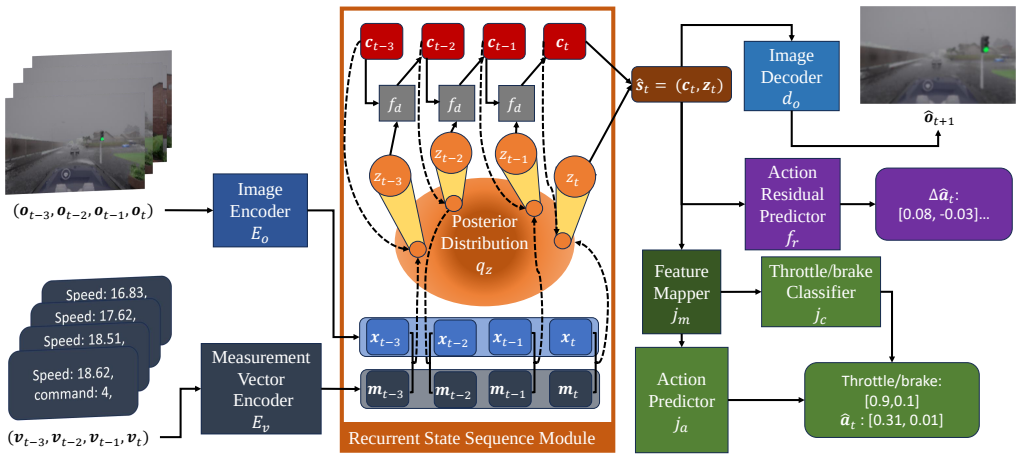

Figure 4: The illustration of proposed framework in 3CIL, with observation history perception window length l set to 3 as example. Dashed edges denote the sampling process. The representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is composed of an image encoder E_o, a measurement vector encoder E_v, and a recurrent state sequence module (f_d, q_z). The predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ is composed of a feature mapper j_m, a throttle-or-brake classifier j_c and an action predictor j_a.

practice, we use the Mean-Squared Error (MSE) loss to carry out the maximum likelihood estimation for future image reconstruction part in Eq.1. MSE loss is also used in optimizing the action residual prediction accuracy, as shown in Eq.2.

Similar to approaches (Cultrera et al., 2023) that divide the action command prediction task into multi-stages to restrain the inertia problem, our predictor model $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ is composed of a throttle-or-brake classifier j_c and an action predictor j_a.

The classifier $j_c(P(go), 1-P(go)|\hat{\boldsymbol{s}})$ processes the coarse action command corresponding to go and stop as a binary classification task, $P(go)$ denotes the probability of increasing speed, while $1-P(go)$ represents the probability of slowing down.

The prediction of j_c is then fed into j_a as part of the action predictor's input features. $j_a(\hat{\boldsymbol{a}}|P(go), 1-P(go), \hat{\boldsymbol{s}})$ predicts the steering angle and the absolute value of acceleration $[\hat{a}_{steer}, \text{abs}(\hat{a}_{acc})]$, the final prediction of $J(\hat{\boldsymbol{a}}|\hat{\boldsymbol{s}})$ is then: if $P(go) > 0.5$, $\hat{\boldsymbol{a}} = [\hat{a}_{steer}, \text{abs}(\hat{a}_{acc})]$, else $\hat{\boldsymbol{a}} = [\hat{a}_{steer}, -\text{abs}(\hat{a}_{acc})]$. We use binary cross-entropy loss to optimize j_c, and MSE loss to optimize j_a. These two loss terms are then multiplied with the sample-weighting term Eq. 4 to produce the final action loss.

A.2.2 SPECS

We list the implementation details of modules in 3CIL in this section. Table 2 concludes the structures of data and major components of 3CIL.

Starting with the representation model $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$, the image encoder $E_o(\boldsymbol{x}|\boldsymbol{o})$ is implemented as a pre-trained ResNet18 model (He et al., 2016), while the measurement vector encoder $E_v(\boldsymbol{m}|\boldsymbol{v})$ is a multi-layer perceptron (MLP) coupled with embedding layers that process discrete navigation commands. $v_{route\_command}$ and $v_{route\_command\_nex}$ are processed through an embedding layer $(\text{embedding\_num} = 7, \text{embedding\_dim} = 8)$, then concatenate with v_{speed} and feed to a MLP $(\text{linear layers} = 3, \text{hidden\_units} = 128, \text{output\_dim} = 128, \text{activiation} = \text{ReLU}(\cdot))$ to produce the encoded feature $\boldsymbol{m}$.

In implementation, the RSSM model of $G(\hat{\boldsymbol{s}}|\boldsymbol{h})$ is composed of a linear layer that maps the feature $\boldsymbol{x}_{t-l:t} \oplus \boldsymbol{m}_{t-l:t}$ extracted from E_o and E_v, into a vector with size = 256, a GRU module whose both input size and hidden size set to 256 is added as the instance of recurrent network f_d, a MLP $(\text{linear layers} = 3, \text{hidden\_units} = 256 + 128, \text{output\_dim} = 128 \times 2, \text{activiation} = \text{ReLU}(\cdot))$ is used to predict the mean and standard deviation of the posterior distribution $q_z(\boldsymbol{z}_t|\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t)$. Finally, a

Table 2: Structures of data and models.

Type		Specification
Observation		$\boldsymbol{o}_t$ = (channel=RGB, width= 240px, height=150px)
Condition		$\boldsymbol{v}_t$ = ($v_{speed} \sim [0, 100]$, $v_{route\_command} \in \{-1, ..., 6\}$, $v_{route\_command\_next} \in \{-1, ..., 6\}$)
Input History		$\boldsymbol{h}_t = (\boldsymbol{o}_{t-l:t}, \boldsymbol{v}_{t-l:t}), l = 4$
Representation Model $G(\hat{\boldsymbol{s}}\|\boldsymbol{h})$	Image Encoder	$E_o(\boldsymbol{x}\|\boldsymbol{o})$: pre-trained ResNet18
	Measurement Vector Encoder	$E_v(\boldsymbol{m}\|\boldsymbol{v})$: MLP(linear layers = 3, hidden_units = 128, output_dim = 128, activiation = ReLU($\cdot$))
	(RSSM) Deterministic State Model	$f_d(\boldsymbol{c}_t\|\boldsymbol{c}_{t-1}, \boldsymbol{z}_{t-1})$: GRU (input_size = 256, hidden_size = 256)
	(RSSM) Stochastic State Model	$q_z(\boldsymbol{z}_t\|\boldsymbol{c}_t, \boldsymbol{m}_t, \boldsymbol{x}_t)$: (mean, std) ~ MLP (linear layers = 3, hidden_units = 256 + 128, output_dim = 128 × 2, activiation = ReLU($\cdot$))
	Extracted Representation	$\hat{\boldsymbol{s}}_t = \boldsymbol{c}_t \oplus \boldsymbol{z}_t$ = Tensor(shape : $[1, 128 + 256]$)
Additional Modules	Action Residual Predictor	$f_r(\Delta\hat{\boldsymbol{a}}_t\|\hat{\boldsymbol{s}}_t)$: MLP (linear layers = 3, hidden_units = 256, output_dim = 2, activiation = ReLU($\cdot$))
	Image Decoder	$d_o(\hat{\boldsymbol{o}}_{t+1}\|\hat{\boldsymbol{s}}_t)$: 3 ConvTranspose2d layers with activiation = ReLU($\cdot$))
Predictor Model $J(\hat{\boldsymbol{a}}\|\hat{\boldsymbol{s}})$	Feature Mapper	j_m : MLP (linear layers = 3, hidden_units = 512, output_dim = 512, activiation = ReLU($\cdot$))
	Throttle/brake Classifier	j_c : MLP (linear layers = 4, hidden_units = 512, output_dim = 2, activiation = ReLU($\cdot$)), and Sigmoid($\cdot$) as output transform function
	Action Predictor	j_a : MLP(linear layers = 4, hidden_units = 512, output_dim = 2, activiation = ReLU($\cdot$))
	Output Action	$\hat{\boldsymbol{a}}_t = [\hat{a}_{steer,t}, \hat{a}_{acc,t}]$ = Tensor(shape : $[1, 2]$)

linear layer connects the computed last historical information vector $\boldsymbol{c}_t$ in timestep t and the sampled hidden state $\boldsymbol{z}_t$, and maps them into $\hat{\boldsymbol{s}}_t$, a feature vector with length = 384.

The feature mapper j_m is a MLP as (linear layers = 3, hidden_units = 512, output_dim = 512, activiation = ReLU($\cdot$)), which processes features that feed into j_c, j_a. The throttle/brake classifier is a MLP as (linear layers = 4, hidden_units = 512, output_dim = 2, activiation = ReLU($\cdot$)), with a Sigmoid($\cdot$) transform the prediction in range $(0, 1)$. The action predictor j_a receives outputs from j_m and j_c, computes its prediction through a MLP with (linear layers = 4, hidden_units = 512, output_dim = 2, activiation = ReLU($\cdot$)).

A.2.3 Training

For 3CIL and baselines we used for comparison that can conduct representation learning and policy learning in separate stages (i.e., DIGIC, PALR, Premier-TACO, RNC, and RAP in Section 4.1), we first conduct their corresponding representation learning with expert demonstrations to obtain stable representation models. For these methods, we use an Adam optimizer (Kingma, 2014) with a learning rate set to $5e{-}6$ to optimize their representation models. An early-stopping monitor module is also added to prevent models from overfitting the training set, with an evaluation set $\mathbf{D_v}$ divided from the training set $\mathbf{D_e}$ in a dividing factor 10%. After training an epoch on the training set, the representation model is required to run on $\mathbf{D_v}$. If the performance increment in $\mathbf{D_v}$ is lower than the optimization threshold $1e-3$, this optimization epoch is marked as a potential invalid update. When the consecutive invalid updates that a representation model encountered have reached the early-stopping threshold (set to 10 in this phase), the optimization process of the representation is finished.

When the training process for representation models was finished, we fixed these models' weights and deployed them in the following policy training phase. Similar to the former stage, an Adam optimizer with a learning rate set to $1e-6$, and an early-stopping monitor with an optimization threshold of $1e-6$ and an early-stopping threshold of 20 are used.

During the representation learning phase of (3CIL, DIGIC, PALR, Premier-TACO, RNC, and RAP) and overall optimization of CIL, data augmentation operations are applied on the imitator's image observations to increase the diversity of the dataset and enhance the robustness of methods. Added data augmentation operations and their corresponding probability are: horizontal flip with probability = 0.3, color jitter (brightness = 0.4, contrast= 0.4, saturation = 0.4 and hue = 0.1) with probability = 0.4, and gray-scale with probability = 0.2.

A.3 Experimental Details

A.3.1 Baselines

As described in Section 4.1, we have picked and implemented baselines including CIL, Keyframe (Wen et al., 2021), DIGIC (Chen et al., 2024), PALR (Seo et al., 2023), Premier-TACO (Zheng et al., 2024), RNC (Zha et al., 2023), and RAP (Chuang et al., 2022) in our experiments. We now introduce each baseline and our implementations.

The Conditional Imitation Learning (**CIL**) approach (Codevilla et al., 2018) eases the complex vision-based driving task by introducing conditions (i.e., the expert's intention, often expressed as route commands) to the model's input. The vanilla CIL pipeline operates in a supervised learning manner by directly minimizing the prediction difference between the policy and expert demonstrations, as: $\mathcal{L}_J = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_a(\boldsymbol{a}_t, J(\boldsymbol{o}_t, \boldsymbol{v}_t))$. In our implementation, we replace the input tuple $(\boldsymbol{o}_t, \boldsymbol{v}_t)$ with $\hat{\boldsymbol{s}}_t \sim G(\hat{\boldsymbol{s}}_t|\boldsymbol{h}_t)$, allowing the representation to capture more information from the temporal aspect. The overall model is still optimized by only the prediction loss.

The Keyframe-Focused Visual Imitation Learning (**Keyframe**) approach (Wen et al., 2021) alleviates the copycat problem by introducing sample-weighting strategies, based on precomputed "action prediction error" (APE) between a copycat policy and the expert demonstrations. Concretely, a copycat policy $\pi_c(\hat{\boldsymbol{a}}|\boldsymbol{a}_{t-l:t-1})$ is trained to use only previous actions $\boldsymbol{a}_{t-l:t-1}$ to predict current action $\hat{\boldsymbol{a}}$. The policy π_c is then used to locate frames that are more likely to be changepoints, by identifying the samples with high APEs. Higher weights are then assigned to these identified samples, regularizing the imitator to focus on changepoints. In our implementation, we train a copycat policy whose

structure and input data are set similar to Wen et al., and use their step$(\cdot)$ function to map samples' APEs into weights on samples. The samples' weights are then plugged into the learning process of CIL.

The Domain Generalizable Imitation Learning by Causal Discovery (**DIGIC**) (Chen et al., 2024) framework, may stand as a representative for approaches that combine causal discovery and IL. Specifically, they picked the covariates that directly contribute to expert action $\boldsymbol{a}_t$ as the input of the downstream predictor. In our implementation, the causal discovery is operated upon the extracted features from the representation model $G(\hat{\boldsymbol{s}}_i|\boldsymbol{h}_i)$. The representation model is first trained with the image reconstruction and posterior regularization losses, to ensure it captures rich information from raw history observations while achieving feature compression. The causal discovery task is then conducted with the mutual information regression test which is provided by the causal discovery toolbox (Kalainathan et al., 2020). Only features that exhibit test statistics that are higher than a threshold(0.20) are picked as input for the downstream BC predictor.

The Past Action Leakage Regularization (**PALR**) method (Seo et al., 2023) bypasses the causal confusion problem with a regularization on the conditional dependence between extracted representation $\hat{\boldsymbol{s}}_t$ and expert's previous action $\boldsymbol{a}_{t-1}$, given current expert's action $\boldsymbol{a}_t$, as $\mathcal{L}_{\text{reg}}(\hat{\boldsymbol{s}}_t; \boldsymbol{a}_{t-1}, \boldsymbol{a}_t)$. Following their work, we adopt the Hilbert-Schmidt conditional independence criterion (HSCIC) from (Park & Muandet, 2020) to perform past action leakage regularization in a non-parametric manner, as: $\mathcal{L}_{\text{reg-HSCIC}}(\hat{\boldsymbol{s}}_t; \boldsymbol{a}_{t-1}, \boldsymbol{a}_t)$ = HSCIC$(\boldsymbol{s}_t, \boldsymbol{a}_{t-1}|\boldsymbol{a}_t)$. Such regularization term is incorporated in the representation learning stage in our implementation, with parameters set as $ridge\_lambda = 1e-3$ and $reg\_coef = 0.1$.

The **Premier-TACO** framework (Zheng et al., 2024) employs a temporal action-driven contrastive loss function for visual representation pretraining, with a new negative example selecting strategy. For a state $\hat{\boldsymbol{s}}_t$, its corresponding positive example is $\hat{\boldsymbol{s}}_{t+k}$, while its negative examples are then selected based from a window with size w centered at state $\hat{\boldsymbol{s}}_{t+k}$ within the same episode, as $\hat{\boldsymbol{s}}_{t,\text{neg}} \sim (\hat{\boldsymbol{s}}_{t+k-w}, ..., \hat{\boldsymbol{s}}_{t+k-1}, \hat{\boldsymbol{s}}_{t+k+1}, ..., \hat{\boldsymbol{s}}_{t+k+w})$. We incorporate this framework in our representation learning stage with positive stride $k = 4$ and window size $w = 5$. The action encoder utilized in Premier-TACO is implemented with a three-layer MLP with 256 hidden units.

Different from Premier-TACO that constructs positive pairs and negative pairs based on temporal indexes, the Rank-N-Contrast framework (**RNC**) (Zha et al., 2023) conducts supervised contrastive learning to shape a robust representation space with guidance from samples' continuous labels. As part of optimization target in our 3CIL method, the $\mathcal{L}_{\text{RNC}}$ in Eq 3 help aligning the distances of samples in the representation space with distances in their labels. To evaluate the benefit from $\mathcal{L}_{\text{RNC}}$ solely, we set RNC as one of our baseline, with implementation as adding it alongside with the image reconstruction and posterior regularization losses.

The Residual Action Prediction (**RAP**) method (Chuang et al., 2022) aims to resolve the copycat problem by designing the residual action prediction objective Eq 2. This approach is also introduced as a baseline, without the sample-weighting strategy in Eq 4.

To alleviate the performance bias incurred by different model capacities, all baselines in our experiments share same architecture design in Appendix A.2.2 and training strategy in A.2.3. Therefore, the major differences in metrics will come from methods' designs.

A.3.2 PLATFORMS

All models used in experiments was trained on a batch size of 64 on a workstation with a RTX4090 GPU. In the testing phase, these models are deployed to the CARLA simulator (version 0.9.12) on another workstation with a RTX3080 GPU.

Table 3 lists configurations for the CARLA simulator used in our experiments, both collecting training data and evaluating models. Specifically, the history subsample frequency is set lower than the actual interaction frequency, as the dynamics in urban driving environments do not contain many high frequency components. Similarly, the frequency of computing and reporting metrics (reward, collision) is set to 4Hz.

Table 3: Configurations of CARLA simulator in experiments.

Configuration	Value
System platform	Windows 10
Graphics quality	quality-level=Epic
Interaction frequency	20Hz
History subsample frequency	4Hz
Perception window length	$l = 4$
Metrics computation frequency	4Hz

A.3.3 Environmental parameters

In the conducted experiments, we modified the weather condition, traffic density, and camera parameter for each scenario used in the evaluation, we listed the environmental parameters used in expert demonstrations and evaluation process in Table 4, and introduce the effect of changing these parameters as follows.

Table 4: Environmental parameters in the training set and test stage.

Environmental parameters	Training set	Test stage
Towns	Scenario 1, 2, 3, 4 (Town01, Town03, Town04, Town06)	Scenario 1, 2, 3, 4, 5, 6 (Town01, Town03, Town04, Town06, Town02, Town05)
Weather group	'ClearNoon', 'WetNoon', 'HardRainNoon', 'ClearSunset'	'WetCloudyNoon', 'SoftRainSunset', 'WetSunset', 'HardRainSunset'
Number of vehicles	Scenario 1: [80, 160], Scenario 2: [40, 100], Scenario 3: [100, 200], Scenario 4: [80, 160]	Scenario 1: 120, Scenario 2: 70, Scenario 3: 200, Scenario 4: 120, Scenario 5: 70, Scenario 6: 120
Front camera FOV	Scenario 1: 70, Scenario 2: 80, Scenario 3: 100, Scenario 4: 120	Scenario 1: 75, Scenario 2: 105, Scenario 3: 95, Scenario 4: 85, Scenario 5: 90, Scenario 6: 110

Except for four scenarios that appeared in the training set, two new scenarios are also included in the test stage. As the $v_{route\_command}, v_{route\_command\_next}$ are offered as navigation information, driving in unfamiliar towns is less terrifying. Still, the introduced new scenarios can examine the applicability of learned patterns of each imitator in new domains.

The weather condition is set differently in the training set and test stage as shown in Table 4, no weather condition in the test stage has been introduced to the imitator in the training dataset. Although the weather in the CARLA simulator does not affect the vehicle's physics, it does affect the lighting condition and the visibility of visual-based imitators, new weather conditions impose trials on imitators' ability in generalization. Moreover, the distribution of weather conditions in the test stage is shifted: sunsets and rain conditions more frequently appear, making the driving task more difficult.

We modify the traffic density by setting the number of other vehicles. Naturally, denser traffic leads to harder challenges on imitators' strategies. In particular, we increase the vehicle count to 200 in Scenario 3, corresponding to Town04 in CARLA, which is a small town. This contradiction between compact map size and heavy traffic load poses stress on imitators and leads to their highest collision rates in the design experiments.

Finally, the camera parameter we modified is the field of view (FOV) of the RGB camera that produces the image observation for imitators. Higher FOV means an imitator can perceive information with a wider perception range, but also brings distortions to objects in an image's corner. Moreover, an object located in an identical spot will be depicted in different sizes when setting FOV at different

levels. Such variances in observation further intensify the extent of distribution shift in the test stage and pose threats to the imitators' generalization.

Figure 5 shows samples from scenarios with environmental parameters set differently.

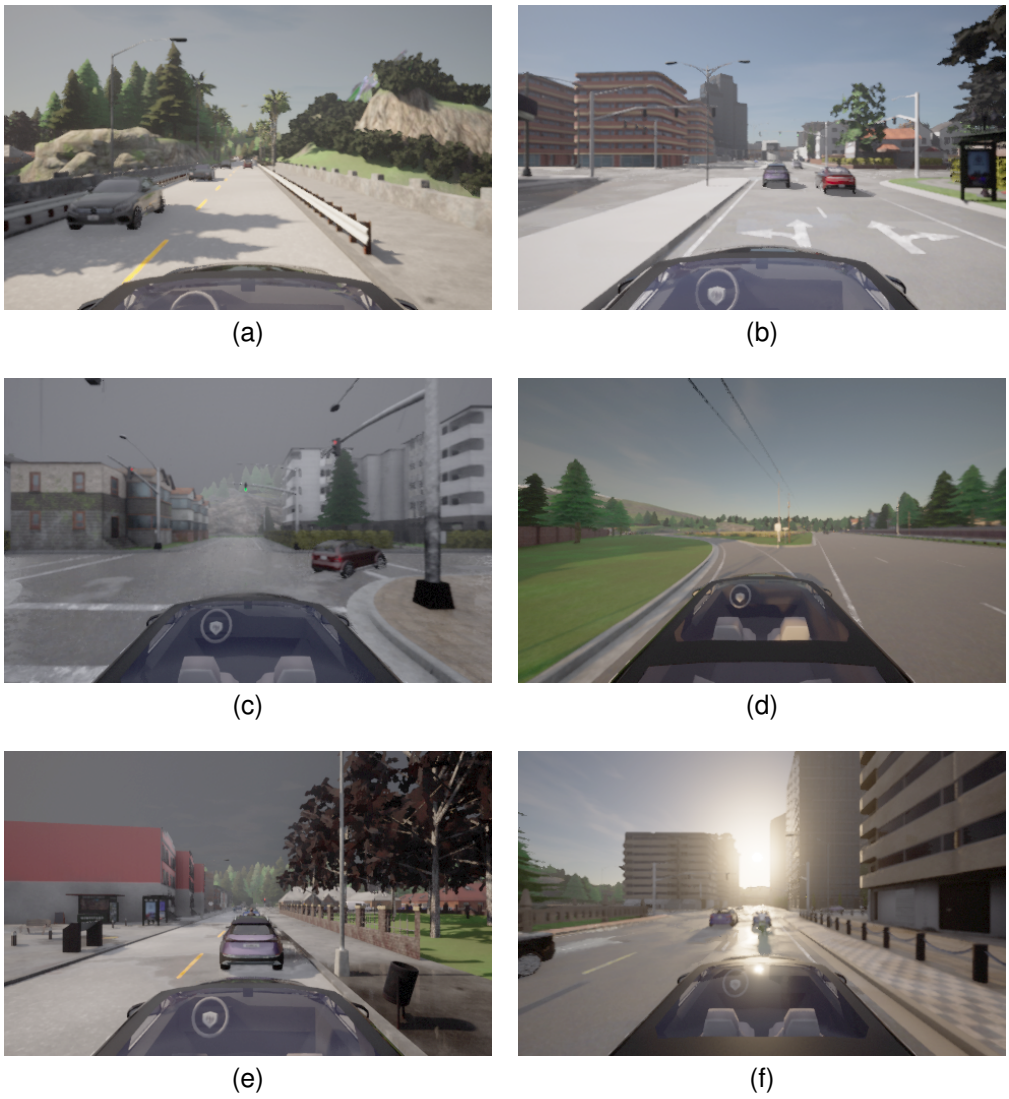

Figure 5: Illustrations of six scenarios used in our experiments. (a): Scenario 1 with the weather set to 'ClearNoon' and camera FOV set to 70. (b) Scenario 2 with the weather set to 'WetNoon' and camera FOV set to 90. (c) Scenario 3 with the weather set to 'HardRainNoon' and camera FOV set to 80.(d): Scenario 4 with the weather set to 'ClearSunset' and camera FOV set to 100. (b) Scenario 5 with the weather set to 'WetCloudyNoon' and camera FOV set to 110. (c) Scenario 6 with the weather set to 'WetSunset' and camera FOV set to 120.

A.3.4 Test Suites

During evaluation, an imitator is required to drive through multiple preset routes in each scenario. A run corresponding to a route is terminated when the imitator: reaches the destination, runs out of the time limit, has a collision with other objects, or is stuck in a place for a while.

Concretely, we picked 10 routes for Scenario 1, 20 routes for Scenario 2, 20 routes for Scenario 3, 6 routes for Scenario 4, 10 routes for Scenario 5, and 10 routes for Scenario 6. In the test stage,

an imitator needs to drive in a route four times, corresponding to the set four weather conditions in Table 4. We set the run time limit to 2000 timesteps, and the stuck detection period is set to 600.

A.3.5 REWARD DESIGN

The reward function is organized as: $R = r_{speed} + r_{position} + r_{rotation} + r_{action}$.

In a certain timestep i, r_{speed} computes the speed reward signal based on the difference between the imitator's current speed $v_{speed,i}$ and desire speed r_{desire_speed}. The desired speed r_{desire_speed} varies when the imitator is around different kinds of objects, set the maximum speed limit for imitator as $maximum\_speed = 30$, the r_{desire_speed} is computed as

$$r_{desire\_speed} = \min(maximum\_speed, maximum\_speed \times vehicle\_factor, \\ maximum\_speed \times light\_factor, maximum\_speed \times sign\_factor), \tag{5}$$

where $vehicle\_factor, light\_factor, sign\_factor$ are modified from the code of Zhang et al. (2021b). Take $vehicle\_factor$ as an example: it first locates the nearest hazard vehicle in the ego vehicle's local coordinate as $\boldsymbol{loc\_veh}$, and computes the distance $dist\_veh = \max(0, \|\boldsymbol{loc\_veh}\|_2 - b_{veh})$ with base distance $b_{veh} = 8$, then runs through a bounding function as $vehicle\_factor = \text{bound}(dist\_veh, 0, 5)/5$. $light\_factor, sign\_factor$ are computed similarly except the different base distances $b_{light} = 6$ and $b_{sign} = 5$. The speed reward is then computed as:

$$r_{speed} = 1 - \frac{|v_{speed,i} - r_{desire\_speed}|}{maximum\_speed}. \tag{6}$$

The position signal $r_{position}$ is computed based on the imitator's lateral distance $d_{lateral}$ with navigation point: $r_{position} = -1 \times (d_{lateral}/2)$.

The rotation punishment $r_{rotation}$ is computed based on the rotation yaw angle differences between the imitator and the navigation point d_{yaw}, as: $r_{rotation} = -1 \times \text{deg2rad}(\text{abs}((d_{yaw} + 180)\%360 - 180))$, where $\text{deg2rad}(\cdot)$ is the function converts angles from degrees to radians.

The r_{action} signal punishes the imitator with $r_{action} = -0.1$ if $|a_{steer,t} - a_{steer,t-1}| > 0.01$ else $r_{action} = 0$.

A.4 ADDITIONAL RESULTS

In this section, we present an empirical test on assumptions made by 3CIL, and a further ablation study on the utilities of each module in 3CIL. We select three representative scenarios from Table4 to conduct experiments on: Scenario 1, Scenario 5, and Scenario 6.

A.4.1 EFFECTS OF HISTORY

An assumption adopted by our work, and previous works that belong to Behavioral Cloning from Observation Histories (BCOH) or POMDP-related approaches is: that using only the most recent frame $\boldsymbol{o}_t$ cannot provide enough essential information for agents to recover an expected policy. Therefore, it is common to design policies that utilize observation history, such as expanding the temporal perceived range of models and using suitable networks for capturing temporal dependency.

To verify whether history helped these approaches capture more crucial information, we designed an intervention analysis similar to (Chuang et al., 2022). Specifically, we replace the original history $\boldsymbol{h}_t = (\boldsymbol{o}_{t-l:t}, \boldsymbol{v}_{t-l:t})$ with the counterfactual history: $[\text{repeat}((\boldsymbol{o}_t, \boldsymbol{v}_t), l)]$, which is replacing all frames in the history with current frame $(\boldsymbol{o}_t, \boldsymbol{v}_t)$. Figure 6 provides examples of the factual setting and the counterfactual history setting. The average performances of models that deployed in experiments under counterfactual history setting are recorded, and compared to their performance under the setting of original history.

Therefore, the difference between performances in these two settings can be seen as the effect of incorporating observation history. We report the results of three methods (CIL, Premier_TACO) in three scenarios, as Table 5. Clearly, for all methods, replacing the original history with the counterfactual history will incur performance degeneration in most Scenarios. Such a phenomenon

Figure 6: An illustration of the factual history setting and the counterfactual history setting.

Table 5: The extent of degeneration in performance when switching to the counterfactual history setting. The ratios of reward dropping and collision rate increasing are computed with the methods' performance in the factual setting (Table 1).

Metric \ Method		CIL	Premier_TACO	3CIL
Scenario 1	Dropped R (%)↓	35.09 (331 → 215)	13.64 (470 → **406**)	28.82 (521 → 371)
	Increased C (%) ↓	12.12 (0.66 → 0.74)	-11.76 (0.85 → 0.75)	29.63 (0.54 → **0.70**)
Scenario 5	Dropped R (%) ↓	685.51(7 → −42)	33.01 (517 → 346)	26.44 (539 → **396**)
	Increased C (%) ↓	20.68 (0.29 → **0.35**)	59.46 (0.37 → 0.59)	22.92 (0.48 → 0.59)
Scenario 6	Dropped R (%) ↓	220.53 (46 → −55)	38.95 (331 → 203)	30.34 (447 → **312**)
	Increased C (%) ↓	61.76(0.34 → **0.55**)	35.29 (0.68 → 0.92)	16.95(0.59 → 0.69)

suggests that models can learn patterns from transitions in observations, and observations in the past will contribute to the prediction quality.

Besides, as proposed in (Chuang et al., 2022), the performance under the counterfactual history setting can also reflect the models' capability in severe copycat status: frozen observations suggest the vehicle is in the stationary state, introduce more evident spurious correlations between current action $\hat{a}_t$ and previous actions $\hat{a}_{t-n:t-1}$. With this insight, we further investigate the degeneration extent of each method.

Interestingly, the ranks of performance under the counterfactual setting (pointed by →) of methods: CIL > 3CIL > Premier_TACO in collision rate, 3CIL > Premier_TACO > CIL in reward, are roughly aligned with their original performance ranks in Table 1. Although the fixed observations in history prevent the models from inferring further information as well as introduce severe causal confusion, still our 3CIL approach manages to achieve relatively less degeneration. This may attributed to the sample-weighting strategy which is utilized in the policy learning phase of 3CIL: as the representation model failed to capture the variations within history from the frozen observations, such a deviation from learned patterns is akin to the samples with high $\boldsymbol{weight}_t$ due to failures in action residual prediction, which belongs to the circumstances we emphasis the 3CIL model to learn.

A.4.2 Sample-weighting Strategy

To showcase the process of our proposed sample-weighting strategy in 3CIL, we pick a circumstance where a sample's $\boldsymbol{weight}_t$ is computed with high value, as shown in Figure 7. The high deviation in action residual prediction is then translated to emphasis on learning this sample.

Previous work also incorporates sample-weighting process, such as (Wen et al., 2021) used two distinct functions to map the APE into samples' weights: step$(\cdot)$ and softmax$(\cdot)$. The step$(\cdot)$ sorts those samples whose APEs are greater than a large proportion of overall samples' APEs (the top 10% samples measured in APE), and assigns these samples with a constant weight W (set to 5.0 in their experiments), other samples are assigned with weight 1.0. Another implementation computes the samples' corresponding weights within a batch, by feeding their APEs into a softmax$(\cdot)$ function.

To evaluate the effectiveness of each sample-weighting strategy, we conduct experiments by changing the weighting process in 3CIL, which results in 3 different performance statistics in Table 6. Specifically, we replace the measurement of error in (Wen et al., 2021) (APE) with the error in

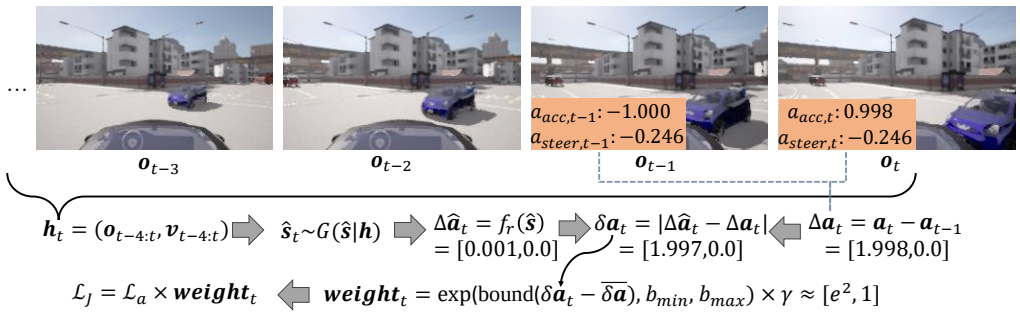

Figure 7: An illustration of the sample with high $weight_t$. For concision, o_{t-4} and $v_{t-4:t}$ are omitted here. Based on features $\hat{s}_t$ obtained from the observation history h_t, the action residual predictor f_r gives its prediction as $\Delta\hat{a}_t = [\Delta\hat{a}_{acc,t} = 0.002, \Delta\hat{a}_{steer,t} = 0.0]$ since it might be safer to remain still, given that the blue car is getting closer. However, the expert chose to accelerate to finish its left turn and give space for the upcoming blue vehicle, which led the actual residual to be: $\Delta a_t = [\Delta a_{acc,t} = 0.001, \Delta a_{steer,t} = 0.0]$. This huge disagreement $\delta a_t = [1.997, 0.0]$ results in a high sample $weight_t$, urging the predictor model to focus more on such an abnormal scene.

Table 6: The extent of degeneration in performance when replacing the sample-weighting strategy in 3CIL with None (no applying weights), step(·) and softmax(·). The ratios of reward dropping and collision rate increasing are computed with the original performance of 3CIL (Table 1).

Metric \ Strategy		None	step(proportion = 20%, weight = 3.0)	softmax(temperature = 0.2)
Scenario 1	R ↑	**491.77** (5.66% ↓)	446.47 (14.3% ↓)	389.97 (25.19% ↓)
	C ↓	0.61 (12.96% ↓)	0.59 (9.26% ↓)	**0.57**(5.56% ↓)
Scenario 5	R ↑	**460.35**(14.51% ↓)	338.75 (37.09% ↓)	402.22(25.31% ↓)
	C ↓	**0.52** (8.33% ↓)	**0.52** (8.33% ↓)	0.54 (12.50% ↓)
Scenario 6	R ↑	**401.51** (10.22% ↓)	359.62 (19.59% ↓)	383.05 (14.35% ↓)
	C ↓	0.71 (20.33% ↓)	0.68(15.25% ↓)	**0.64** (8.47% ↓)

residual prediction. The residual prediction receives the extracted features $\hat{s}_t$ as input and can identify more general abnormal scenes beyond the copycat problem.

Out of the blue, the incorporation of sample-weighting processes does not always come with benefit (compared to "None", in the measurement of Reward). This may be caused by the potential mismatch between the functions' hyper-parameters and training data distribution, as these functions are rather sensitive to the choice of hyper-parameters: the step$(\cdot)$ especially, which shows extremely low average speed (3.55 in Scenario 1) when set hyper-parameters as default. We expect the performance of these sample-weighting strategies can be further improve when finetuning them with domain knowledge. Still, both step$(\cdot)$ and softmax$(\cdot)$ generally reduce the collision rate of the imitator, which may be attributed to the up-weighting on potential changepoints.

In conclusion, the choice of sample-weighting strategy is flexible, as long as the function that computes weights is a monotonic non-decreasing function of the action residual prediction error, similar to the setting of (Wen et al., 2021). However, it is essential to tune the function to achieve a balance between overly flat (not enough emphasis on abnormal scenes) and overly steep (potentially underfitting with ordinary driving scenes).

A.4.3 Effectiveness of each design

Table 7: Ablation studies.

Metric \ Method		No $\mathcal{L}_{\text{ar}}$, no $\boldsymbol{weight}_t$	No $\mathcal{L}_{\text{RNC}}$, no $\boldsymbol{weight}_t$	No $\mathcal{L}_{\text{RNC}}$	No $\boldsymbol{weight}_t$	3CIL
Scenario 1	R ↑	411.31	383.54	476.35	491.77	**521.26**
	C ↓	0.55	0.60	0.57	0.61	**0.54**
Scenario 5	R ↑	299.72	302.50	387.24	460.35	**538.50**
	C ↓	0.50	0.59	0.55	0.52	**0.48**
Scenario 6	R ↑	**447.44**	195.53	234.44	401.51	447.24
	C ↓	0.64	0.63	0.63	0.71	**0.59**

We examine the effect of each design decision in our approach and present the results in Table 7.

The results show that all major designs in 3CIL have contributed to the overall performance. Concretely, the supervised contrastive learning loss $\mathcal{L}_{\text{RNC}}$ ("No $\mathcal{L}_{\text{ar}}$, no $\boldsymbol{weight}_t$") provides important guidance in arranging the representation space, helps the imitator to achieve better alignment with the expert policy. While the presence of action residual prediction task $\mathcal{L}_{ar}$ enhances the imitator's capability in capturing crucial influence from previous actions $\boldsymbol{a}_{t-1}$, the $\mathcal{L}_{ar}$ stands alone ("No $\mathcal{L}_{\text{RNC}}$, no $\boldsymbol{weight}_t$") does not provide satisfactory gains. After incorporating the weighting strategy, the performance of an imitator ("No $\mathcal{L}_{\text{RNC}}$") does show significant improvement, but still fails to generalize in some cases, which emphasizes the importance of contrastive learning in shaping a robust representation space. Finally, the divergence between "No $\boldsymbol{weight}_t$" and 3CIL provides the evidence that adjusting weights on diverse samples is beneficial, as it can guide the imitator to focus on crucial changepoints and abnormal scenes.