Causal Imitative Models for Autonomous Driving

Mohammad Reza Samsami
Sharif University of Technology
Tehran, Iran
MohammadRezaSamsami76@gmail.com

Mohammadhossein Bahari
École polytechnique fédérale de Lausanne
Lausanne, Switzerland
mohammadhossein.bahari@epfl.ch

Saber Salehkaleybar
Sharif University of Technology
Tehran, Iran
saleh@sharif.edu

Alexandre Alahi
École polytechnique fédérale de Lausanne
Lausanne, Switzerland
alexandre.alahi@epfl.ch

Abstract

Imitation learning is a powerful approach to leverage data from demonstrations of expert driver to learn autonomous driving policy. In particular, behavior cloning reduces policy learning to supervised learning by training a function that maps states to expert actions. However, driving policies trained via behavior cloning exhibit undesirable behaviors since they neglect the causal structure of expert demonstrations. In this paper, we study two problems in autonomous driving policies trained with imitation learning: inertia and collision problems. We propose Causal Imitative Model (CIM), a method that explicitly discovers the causal model and utilizes it in training the policy. Specifically, CIM disentangles the input to a set of latent variables, selects the causal variables, and determines the next position by leveraging the selected variables. Our experiments show that our method outperforms previous work in terms of inertia and collision rates. Moreover, thanks to exploiting the causal structure, CIM shrinks the input dimension to only two, hence, can adapt to new environments in a few-shot setting. We will make our code and data publicly available.

1 Introduction

Self-driving car is a promising technology that makes a remarkable revolution in the transportation industry. The technology became more accessible in recent years due to the advancements of deep learning. The core task in a self-driving car is autonomous driving from raw sensory inputs. Imitation learning is a desirable approach for learning the task as it needs off-line demonstrations of the expert in contrast to the need for exploration in reinforcement learning which might put the agent at risk. Moreover, instead of a carefully designed reward function in reinforcement learning, it suffices to use an imitation loss. However, learning the complex task of autonomous driving only by observing inputs and the corresponding outputs in imitation learning is challenging due to the complicated reasoning inherent in the task.

Many researchers opt for studying the task of autonomous driving using imitation learning [28, 8, 7, 9]. The main difficulty in learning driving behavior is to extract the implicit expert driver reasoning over various sensory inputs. Although previous work has achieved successes for learning the task, yet two critical issues exist. First, when the trained agent stops (*e.g.,* at a traffic light), it often stays static. This problem is known as "inertia problem" and has been observed in [7]. Second, the trained agent does not avoid colliding with other cars, known as "collision problem", as shown by our experiments.

Preprint. Under review.

Although the problems are critical, they have not received a concrete study yet. In this work, we propose a causal framework to address these two problems.

Causal learning in imitation learning has been recently studied. In particular, causal learning when sensory inputs of the learner and the expert are different, has been investigated in [33]. In [12], the authors showed the phenomenon of learning causally-incorrect policies with naive imitation learning. They proposed a causal learning algorithm to mitigate this issue; however, the complexity of the algorithm for the high dimensional inputs in the autonomous driving task is prohibitive. In [7], the authors identified learning from spurious correlations rather than real causes as the source of the inertia problem in the trained autonomous driving policies. We address the inertia and collision problems by introducing a causal model for autonomous driving.

We propose *Causal Imitative Model* (CIM) for autonomous driving to mitigate inertia and collision problems. Our method can be applied to the existing imitative models to resolve these two main issues. CIM disentangles the input to a set of latent variables. Then, from the latent variables, it selects the causal variables for determining the next position using Granger causality [10]. Finally, the policy is trained to find appropriate actions leveraging the causal variables. We conduct experiments using CARLA simulator [8] to demonstrate the effectiveness of CIM in reducing both inertia and collision problems comparing with state-of-the-art. We show that CIM finds meaningful latent variables as the causes of speed. Specifically, it learns the task employing only two causal latent variables. Therefore, by leveraging a shallow network, it can adapt to new domains with a few samples. Our main contributions are summarized as follows:

- We introduce CIM, a causally learned imitative model for autonomous driving.
- Our experiments show that CIM learns meaningful causes and reduces inertia and collision problems by more than 50% and 7%, respectively. It also adapts to new domains with few examples.
- We conduct ablation studies to reveal the role of each building block of CIM in the final results.

2 Related work

2.1 Imitation learning

Imitation learning, also known as learning from demonstrations, is a powerful and practical framework in scenarios that are too challenging to provide a clear reward signal and it has been applied to a wide variety of domains [1, 3, 27, 34]. Specifically, in safety-critical tasks like autonomous driving, imitation learning has gained lots of attention, and its first application in autonomous driving dates back to 1989 in [25]. Imitation learning through behavior cloning is a form of supervised learning which has been studied in autonomous driving tasks [2, 3, 6].

The most relevant approaches to our work are the methods proposed in [28, 9]. In [28], the authors developed a trajectory density model to plan future trajectories. Their approach benefits from imitation learning in acquiring desirable behavior and exploiting the flexibility of model-based reinforcement learning to adapt to new tasks. This model has been improved in [9] by using a Bayesian algorithm to do epistemic uncertainty-aware planning, advantageous in making robust decisions in out-of-distribution situations. The limitations of behavior cloning are explored in [7]. They recognized weak performance for rare events due to the bias in datasets, high variance of the learned policy because of stochastic learning and causal confusion due to lack of explicit causal model as main challenges of imitation learning in autonomous driving. Inspired by their work, we propose a causal imitative model to address the lack for a causal model.

2.2 Causal structure learning

Learning causal structures from the observational data is one of the primary problems in statistics and machine learning. Causal relationships among a set of variables are commonly represented by a directed acyclic graph (DAG) where there is a direct link from variable X to variable Y if X is direct cause of Y. In the literature of causality, recovering the true causal graph from observation data has been studied in two main settings: random variables and time series.

In the random variable setting, without considering further assumption on the causal system, one can recover the true causal graph up to a Markov equivalence class by utilizing constraint-based [22, 31] or score-based approaches [21, 5]. In order to uniquely identify the causal graph, it is required to consider additional assumptions on the causal system. For instance, LiNGAM algorithm [30] can return the true causal graph in linear systems with non-Gaussian exogenous noises. Moreover, under some mild assumptions, it has been shown that the true causal graph can be identified in non-linear system with additive exogenous noises [16].

In the setting of time series, most effort was confined to define the statistical definition of causality such as Granger causality [10, 11] or transfer entropy [29]. In 1960's, Granger proposed a definition of causality between random processes. More specifically, in this definition, random process X^t is a Granger-cause of random process Y^t if knowing the past of X^t up to time t, can improve the prediction of future of Y^t. Later, Marko proposed another notion of causality between time series based on directed information (DI) [20]. More recently, inspired by Ganger causality, it has been shown that the causal structure of a dynamical system can be recovered by computing DI from samples of time series [26]. In [23], TiMINO algorithm has been proposed to recover causal structures between time series when the exogenous noises are additive. In [32], by employing multi-layer perceptrons (MLP) and recurrent neural networks (RNN), a class of methods has been proposed to capture non-linear Granger causality in time series. In [13], two non-linear regression methods have been proposed to recover causal structures for a wide class of causal systems in both random variable and time series settings. In this work, we utilize Granger causality [10] to learn causes from time-series data.

2.3 Causal imitation learning

Interest in the intersection of causality and imitation learning has been raised very recently. In [33], the authors focused on the feasibility of imitation from the causal perspective. They explained that when sensory inputs of the expert and the learner vary (*i.e.,* they do not have the same causal model), the implicit assumption that imitating the expert translates into high rewards to the learner is not justified. Therefore, they relaxed this assumption and introduced a graphical criterion determining learner's imitability from demonstration data.

The need for causal modeling in imitation learning was highlighted in [12, 7]. Authors in [12] claimed that unless we do not maintain an explicit causal model, true causes cannot be easily identified from spurious correlations. Therefore, policies learned via behavior cloning can suffer from causal confusion phenomenon. They proposed an algorithm to resolve this issue by learning the correct causal model and mapping it to an optimal policy. However, the complexity of algorithm grows exponentially with the number of features. In [7], the authors observed the existence of inertia problem in autonomous driving policies learned from naively imitating the expert. They recognized the main source of the problem as the tendency of the learned policy towards spurious correlations instead of real causes. Moreover, they proposed to use a speed prediction branch as an auxiliary loss to remedy the problem, however, it does not essentially lead to learn the causal structure. In contrast, we propose a causal learning method to address the inertia problem.

The structure of the rest of the paper is as follows: In Section 3, we formulate the problem of imitation learning in self-driving cars. In Section 4, we present CIM method and explain its different parts. We provide experimental results for CIM in Section 5 and compare it with previous work. Finally, we conclude the paper in Section 6.

3 Problem formulation

In this section, we formalize our imitation learning problem and introduce some preliminaries and definitions. In an imitation learning setting, the learner utilizes demonstrations from an expert, *e.g.,* a human or a rule-based algorithm, to learn a policy for performing a given task.

In each time step t, the agent has access to a high-dimensional observation of the scene around itself denoted by O^t, paired with expert future trajectory $\mathcal{T}^t$. The observations include the information of the static parts of the scene, known as the context, and the dynamic agents in the scene. The expert policy generates the trajectories, $\mathcal{T} \sim \pi(.|O)$, and we aim to approximate π by using $(O^t, \mathcal{T}^t)$ tuples. The simplest form of imitation learning is behavior cloning, which reduces a policy learning

problem to a supervised learning problem. In this approach, the agent learns to map π from O to $\mathcal{T}$ directly. We also assume that the agent has access to an inverse dynamic model as a controller. Controller receives the current and subsequent positions of the car, finds a set of speed values and finally offers low-level control commands A^t, such as braking and steering, required to reach to the desired positions. The agent executes A^t, which updates the environment and provides the next observation.

3.1 The underlying causal mechanism of expert's demonstrations

Inspired by the causal model in [12], we hypothesize that the underlying causal mechanism of expert demonstrations is a complex sequential decision-making process. Every observation O^t is disentangled into some generative factors $Z^t = [Z_1^t, Z_2^t, ..., Z_k^t]$, each representing some information regarding the state of the world where k is the number of generative factors. For instance, one factor can represent the presence of cars in front, or the other can represent traffic light state, or weather condition. We consider Z^t as the state at time step t. To plan the future trajectory $\mathcal{T}^t$, first, the expert infers the speed for the next timestep S^t_{forward} based on the current state Z^t. Then, the expert uses the state and the inferred speed, to induce a trajectory. Note that only some of the factors in the state directly influence the speed variable S^t_{forward} and the rest are nuisance variables. We consider the influential factors as the causes (or parent set) of the speed.

4 Method

We present *Causal Imitative Model* (CIM), a method based on the causal mechanism of expert's demonstrations in Section 3.1 to learn the expert policy. As we explained in the previous section, speed prediction is necessary for planning future trajectory. Moreover, inertia and collision problems are speed-related problems. Hence, we mainly focus on the speed prediction part of the autonomous driving system. CIM consists of two main parts:

- **An imitative model** that deals with context-related navigation in the scene. It estimates the density of the distribution over future expert positions $q(\hat{\mathcal{T}}|O;\theta)$. For this part, we employ an off-the-shelf imitative model and train it using maximum likelihood estimation: $\theta^* = \arg\max_\theta \mathbb{E}\big[\log q(\hat{\mathcal{T}}|O;\theta)\big]$.
- **A modular causal pipeline** that estimates the desired speed and has three consecutive parts. First, there is a *Perception model* that embeds observations into a disentangled representation. Then, the *Cause selector* identifies which dimensions correspond to the direct causes of the speed. *Speed predictor* finds the speed in the next step based on the cause variables. Finally, the Controller adjusts the imitative model's speed by utilizing the predicted speed value. We will describe this pipeline in more detail.

4.1 Perception model

At each time step, we get a high dimensional observation O^t from the environment. As discussed in section 3.1, the expert makes decisions based on the disentangled generative factors Z^t of the observation. Therefore, the role of the *Perception model* is to learn latent variables $\hat{Z}^t$ as a disentangled abstract representation of observed input O^t. In a desired disentangled representation, each latent variable in $\hat{Z}^t$ is sensitive to changes in one generative factor while being relatively invariant to other factors [14]. Variational autoencoders (VAE) [18] are commonly employed to obtain disentangled factors from observations [15, 17, 4]. We implemented and trained a β-VAE [15] as our *Perception model* to compress the pre-processed observations into a disentangled representation of 128 variables. Appendix A provides the architecture and implementation details.

4.2 Causal speed predictor

Having learned the disentangled representation, we identify causes of the next step speed S^t_{forward} from learned latent variables $\hat{Z}^t$. To this end, we consider time series of $\hat{Z}_i, i = 1, \cdots, k$ and take speed as the target variable. Here, we use the notion of Granger causality [10]. In particular, we say

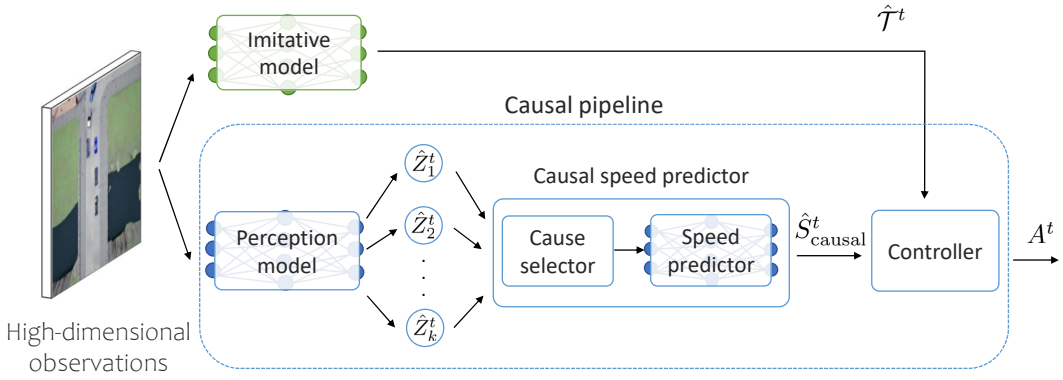


Figure 1: Illustration of our method. Imitative model is responsible for providing a context-based trajectory. CIM deals with the interaction between vehicles and finds the speed of the vehicle. It first disentangles the input into latent variables by *Perception model*. Then, *Cause selector* keeps the causes of the speed. These causes are processed by *Speed predictor* to find the speed. Finally, *Controller* merges the trajectory and the speed and provides control signals.

X^t is a Granger-cause of Y^t if knowing the past of X^t aids in predicting the future of Y^t. More specifically, let $\sigma_Y(h|\Omega^t)$ be the mean square error (MSE) of the h-step predictor of Y^t at time t given information Ω^t. X^t is said to be a Granger-cause of Y^t if:

$$\exists h > 0,\ s.t.\ \sigma_Y(h|\Omega^t) < \sigma_Y(h|\Omega^t \backslash \{X^i\}_{i=0}^t), \tag{1}$$

where Ω^t includes all the information in the universe related to the past or present of Y^t. There are other definitions of Granger causality in form of checking conditional dependency in time or measuring transfer entropy. It can be shown that one can identify the true causes of a target variable using Granger causality if we are observing all the variables that may have influences on it[1]. This assumption is commonly called "causal sufficiency" assumption. By considering this assumption, we are able to perform Granger causality tests in order to recover the causes of speed.

We consider the selected variables from Granger causality tests as the true causes of speed and design *Speed predictor* based on them. We intentionally train *Speed predictor* independently from the *Perception model* to reside representation complexity in *Perception model*. Our experiments showed that for the predictor, it suffices to consider a fully-connected network fed by the concatenated last three steps of cause variables to predict the speed (see details in Appendix B). Thanks to our approach, we came up with two causal variables among 128 latent variables which led to a very dense network beneficial in few-shot adaptation setting. We will investigate this point more in the experiments.

4.3 Controller

In our method, the trajectory planned by the imitative model and the speed predicted by the causal model are fed to *Controller*, which computes the corresponding speed for the trajectory. Next, *Controller* measures the forward target speed using the trajectory's speed and the predicted speed. Then, *Controller* calculates the forward speed error, the difference between the current forward speed and the forward target speed. Based on the speed error, *Controller* uses the control as throttle or brake. Lastly, the agent performs the calculated control commands, making a transition in the environment. Note that *Controller* also calculates other commands, but we do not mention them here as they are irrelevant to the speed.

More specifically, suppose at time t, the planned trajectory's speed is $\hat{S}^t_{\text{trajectory}}$, and the causal predictor returns $\hat{S}^t_{\text{causal}}$. *Controller* computes $\hat{S}^t_{\text{forward}}$ by the following formula:

$$\hat{S}^t_{\text{forward}} \leftarrow (1-\alpha)\hat{S}^t_{\text{trajectory}} + \alpha \hat{S}^t_{\text{causal}}, \tag{2}$$

[1]There are some other mild assumptions that should be taken into account. Please check Theorem 10.3 in [24] for more details.

where α is a constant step-size parameter between 0 and 1. *Controller* computes the forward speed error, $e_s^t = \hat{S}^t_{\text{forward}} - S_{\text{current}}$ where S_{current} is the current speed of the car, and then it computes accelerator control U_s^t using e_s^t. Finally, it calculates the brake and throttle values as follows:

$$\text{throttle} \leftarrow \mathbb{1}(e_s^t > 0) \cdot U_s^t, \text{brake} \leftarrow \mathbb{1}(e_s^t \leq 0) \cdot U_s^t \tag{3}$$

5 Experiments

In this section, we evaluate the performance of CIM under multiple settings. We conduct experiments to address the following points: **(1)** Does the proposed approach reduce collision and inertia events? **(2)** Does our method benefit from merits of causal reasoning such as few-shot domain adaptation? **(3)** How does each part of the method contribute to the results?

5.1 Dataset

We evaluate the methods using CARLA urban driving simulator [8], which is a standard platform for research in autonomous driving and has enabled many researchers to train policies in a dynamic urban environment[2]. CARLA provides various environments with different towns and number of cars. We collected data from `Town01` to train the imitative model and *Perception model*, described in detail in supplementary materials. All the experiments were performed on Nvidia RTX 2080 GPU.

5.2 Performance measures

To study the extent our method is effective in mitigating the inertia and collision problems, we introduce various metrics to measure the percentage of unsuccessful attempts based on inertia and collisions:

Inertia rate: The percentage of navigation tasks that failed due to the inertia problem. Motivated by [7], we consider a task as a failed one due to the inertia problem if the agent has almost zero speed for at least 15 seconds before the timeout.

Collision rate: The percentage of navigation tasks ending up in an accident with another car.

Error rate: Sum of collision rate and inertia rate. Comparing methods based on both inertia and collision rates is challenging as there is a trade-off between these two metrics; to reduce inertia rate, the agent applies more throttle leading to higher speed values; consequently, colliding with the other cars is more likely. Therefore, we need to aggregate collision and inertia rates into one metric. In order to give them equal importance, we considered the sum of them as the error rate to quantify the total failures occur in the tasks. We will use this metric to compare different methods throughout the experiments.

5.3 Baselines

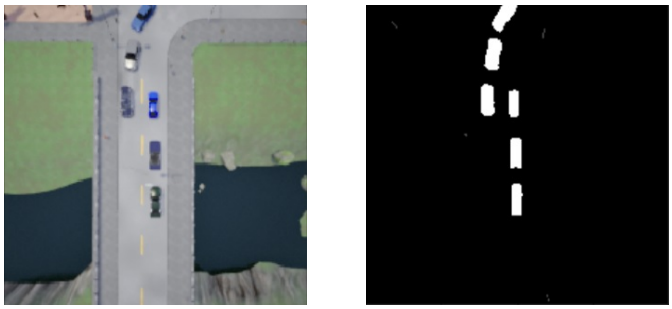
We implemented CIM by leveraging the Deep Imitative Model (**DIM**) [28] as the imitative model. CIM adjusts DIM's speed by a causal speed prediction approach.

To compare CIM's performance with previous work, we considered the two state-of-the-art models, DIM and Robust Imitative Planning (**RIP**) [9] and used their public code for the experiments. Furthermore, as ablation studies, we implemented the following baselines; **CIM-MLP**, which is the same as CIM except that instead of discovering the causes of the speed variable in the time series, it feeds all latent variables into *Speed predictor*; **CIM-entangled** which is identical to CIM except it learns entangled representation by setting $\beta = 1$ in training β-VAE.

5.4 Implementation details

The input images to the model O^t are CARLA bird-view images. Ego car's speed mostly changes due to the interaction with other cars, and context information (*e.g.,* the road shape) has little impact

[2]CARLA specific code is distributed under MIT License. CARLA specific assets are distributed under CC-BY License.



(a) Bird-view RGB observation (b) Pre-processed observation

Figure 2: Pre-processing input observations. We remove context information to avoid feeding speed-irrelevant information to *Perception model*.

on the speed. Hence, in the pre-processing phase, we only keep the information of nearby cars. In particular, we transform bird-view images into gray-scale images, in which, we consider white color for the pixels corresponding to a car and the rest as black (see Figure 2). While this procedure eliminates spurious correlations of context information to cars' speed, still the model can take nuisance information from other agent's positions. For instance, the positions of vehicles on the opposite lane are nuisance information which should be identified by the model. By employing Granger causality as explained in Section 4.2, we distinguish cause variables from nuisance variables associated with cars' interactions.

To disentangle the input, we use a 4 layer convolutional network followed by fully-connected networks as the *Perception model*. It disentangles O^t to 128 latent variables. *Speed predictor* is a two layer feed-forward network with 12 hidden neurons. It is worth mentioning that the training data has 100000 samples.

5.5 Interpreting the discovered cause variables

Among 128 disentangled latent variables, some are causes of speed and the rest are nuisances. After applying Granger causality, we found that only two latent variables have statistically significant effect on speed. By applying tools from [19], we analyzed the characteristics of these two variables. In particular, we plotted latent space's traversals by changing each of these two variables' values one at a time in the range of $[-3, 3]$ while keeping all other latent variables fixed to their inferred values. We observed that traversal of one of the found variables produces *smooth variations in the location of the front car and the back car* of the ego car, and the other one changes *the number of the front cars*, as depicted in Figure 3. Clearly, both selected factors have direct impact on the car's speed.

5.6 Results

To address question **(1)** regarding improvements in inertia and collision rates, we evaluated CIM, DIM, and RIP on 100 tasks in `Town01`. To compare the methods in an environment with distributional shift, we also assessed them on 100 collected tasks from `Town02`. In the experiments, we specified the town, start and destination points, and the number of cars and pedestrians in every task. Any other initial configuration, like locations of other cars, was initialized randomly. We executed three trials per task to reduce the possible influence of randomness in initial states. The results are reported in Table 1. Our method has almost 50% less inertia rate in both towns compared to DIM and RIP. While collision rates are competitive in `Town01`, CIM could reduce DIM's collisions by 16% in `Town02`. It also has 7% less collisions with respect to RIP which is designed for handling out-of-distribution scenarios. The results show the noticeable impact of our causal learning in reducing the error rate in both in-distribution and out-of-distribution data.

5.6.1 Ablation studies

In this part, we report ablation studies to answer questions **(2, 3)**. More specifically, we want to study the importance of different parts of CIM. Additionally, we want to investigate if CIM benefits from

(a) Traversals of the first factor which changes the location of the front and the back cars

(b) Traversals of the second factor which changes the number of the front cars

Figure 3: The result of traversal experiments on two cause variables. Each image shows the result of traversing a cause variable. To create these images, we first infer the latent representation. Next, we traverse a single latent variable while keeping the other latent variables unchanged. The results indicate that these two variables are essential for interacting with other vehicles for inferring ego car's speed.

Table 1: Comparison of different autonomous driving methods on CARLA's navigation tasks in `Town01` and `Town02`.

	`Town01`			`Town02`		
Methods	**Error** ↓	Inertia ↓	Collision ↓	**Error** ↓	Inertia ↓	Collision ↓
DIM [28]	65%	40%	**25%**	74%	30%	44%
RIP [9]	77%	42%	35%	83%	43%	40%
CIM (ours)	**49%**	**21%**	28%	**54%**	**17%**	**37%**

causal learning in the setting of few-shot domain adaptation. To this end, we assessed CIM and its two variants CIM-MLP, and CIM-entangled on `Town01` and `Town02`. The results are provided in Table 2. We discuss the results in the next paragraphs.

Table 2: Assessing the effect of components in CIM.

	`Town01`			`Town02`		
Methods	**Error** ↓	Inertia ↓	Collision ↓	**Error** ↓	Inertia ↓	Collision ↓
CIM-entagled	53%	29%	**24%**	62%	21%	41%
CIM-MLP	**47%**	**17%**	30%	**54%**	**17%**	**37%**
CIM	49%	21%	28%	**54%**	**17%**	**37%**

Effectiveness of disentangled representation. In order to be able to separate causal and nuisance factors, CIM disentangles input to disentangled latent factors. Otherwise, every latent dimension possibly captures both nuisance variables and causes; consequently, discovering the true causes becomes infeasible.

To investigate the impact of disentanglement, we evaluated the performance of CIM-entangled. As shown in Table 2, CIM has better performance in terms of error rate than CIM-entangled in both towns, indicating that disentanglement plays an important role. Note that CIM-entangled has noticeably better performance than DIM, which is probably due to elimination of the context data as nuisance information with respect to the speed.

Explicit causal discovery and few-shot setting. *Perception model* disentangles input to 128 latent variables. Then, *Cause selector* in CIM selects causal latent variables by applying causal structure learning algorithms mentioned in section 4.2. This leads to selecting two variables as the causes of the speed leading to a very shallow network. To show the advantage of this approach, we compared

Table 3: Evaluating the performance of CIM and CIM-MLP in few-shot domain adaptation in `Town03`.

Methods	**Error** ↓	Inertia ↓	Collision ↓
CIM-MLP	38%	22%	16%
CIM	**29%**	**20%**	**9%**

CIM with CIM-MLP. CIM-MLP learns to predict the speed using all 128 latent variables with a much more complex network than CIM and it does not explicitly learn the causal structure. We observed no significant difference in Table 2, indicating fully-trained CIM-MLP learns the causal structure from the data implicitly. However, this might not hold in few-shot settings with limited available data.

Online adaptation of models to new environments is a valuable feature for practical use. Desirably, this adaptation should be achieved with small number of samples. Hence, to study both the capability of models in domain adaptation and their performance in low-data regime, we perform few-shot domain adaptation. In this experiment, first, we fully trained speed predictors CIM and CIM-MLP in `Town01`, and then, tuned them with a low number of samples (100 samples) from another town with a different structure. We selected `Town03` for the test town as in contrast with `Town01`, it is the most complex town containing a 5-lane junction and a roundabout. Table 3 reports the performance of models and shows that CIM outperforms CIM-MLP in all metrics. This indicates CIM-MLP requires more samples to perform as good as CIM, showing the advantage of the employed causal structure learning.

6 Conclusions

In this paper, we proposed Causal Imitative Model (CIM), which can discover the causal model and leverage it to resolve the inertia and collision problems in the autonomous driving task. Experimental results in the CARLA driving simulator validated that our solution successfully recovers causally related concepts, has a better performance than state-of-the-art methods in terms of inertia and collision problems, and can adapt to new environments by observing few samples from them. Our approach can be broadly applied to learning from demonstration problems, and utilizing it in more general settings of imitation learning can be an interesting future research direction.

Multiple unintended consequences could be considered for self-driving technology. First and foremost, many jobs may be lost after utilizing fully functional self-driving cars, constituting a major unemployment crisis. Moreover, insurance companies often consider higher costs for full coverage car insurance based on current risks such as drunk driving incidents. As self-driving cars eliminate such human error, insurance companies will be under pressure to reduce the insurance rates which in turn, may lead to potential bankruptcy among these companies. Lastly, using self-driving cars may expose us to cyber-security attacks and serious measures should be taken into account in this regard.

References

[1] Abbeel, P., Coates, A., Ng, A.Y.: Autonomous helicopter aerobatics through apprenticeship learning. The International Journal of Robotics Research **29**(13), 1608–1639 (2010)

[2] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J., et al.: End to end learning for self-driving cars. arXiv preprint arXiv:1604.07316 (2016)

[3] Chen, C., Seff, A., Kornhauser, A., Xiao, J.: Deepdriving: Learning affordance for direct perception in autonomous driving. In: Proceedings of the IEEE international conference on computer vision. pp. 2722–2730 (2015)

[4] Chen, R.T., Li, X., Grosse, R., Duvenaud, D.: Isolating sources of disentanglement in variational autoencoders. arXiv preprint arXiv:1802.04942 (2018)

[5] Chickering, D.M.: Optimal structure identification with greedy search. Journal of machine learning research **3**(Nov), 507–554 (2002)

[6] Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A.: End-to-end driving via conditional imitation learning. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 4693–4700. IEEE (2018)

[7] Codevilla, F., Santana, E., López, A.M., Gaidon, A.: Exploring the limitations of behavior cloning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9329–9338 (2019)

[8] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017)

[9] Filos, A., Tigkas, P., McAllister, R., Rhinehart, N., Levine, S., Gal, Y.: Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In: International Conference on Machine Learning. pp. 3145–3153. PMLR (2020)

[10] Granger, C.W.J.: Economic processes involving feedback. Information and control **6**(1), 28–48 (1963)

[11] Granger, C.W.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica: Journal of the Econometric Society pp. 424–438 (1969)

[12] de Haan, P., Jayaraman, D., Levine, S.: Causal confusion in imitation learning. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)

[13] Heydari, M.R., Salehkaleybar, S., Zhang, K.: Adversarial orthogonal regression: Two non-linear regressions for causal inference. Neural Networks (2021)

[14] Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., Lerchner, A.: Towards a definition of disentangled representations. arXiv preprint arXiv:1812.02230 (2018)

[15] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework (2016)

[16] Hoyer, P.O., Janzing, D., Mooij, J.M., Peters, J., Schölkopf, B.: Nonlinear causal discovery with additive noise models. In: Advances in neural information processing systems. pp. 689–696 (2009)

[17] Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning. pp. 2649–2658. PMLR (2018)

[18] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[19] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: international conference on machine learning. pp. 4114–4124. PMLR (2019)

[20] Marko, H.: The bidirectional communication theory-a generalization of information theory. IEEE Transactions on communications **21**(12), 1345–1351 (1973)

[21] Meek, C.: Graphical Models: Selecting causal and statistical models. Ph.D. thesis, PhD thesis, Carnegie Mellon University (1997)

[22] Pearl, J.: Causality. Cambridge university press (2009)

[23] Peters, J., Janzing, D., Schölkopf, B.: Causal inference on time series using restricted structural equation models. In: Advances in Neural Information Processing Systems. pp. 154–162 (2013)

[24] Peters, J., Janzing, D., Schölkopf, B.: Elements of causal inference: foundations and learning algorithms. The MIT Press (2017)

[25] Pomerleau, D.A.: Alvinn: An autonomous land vehicle in a neural network. Tech. rep., Carnegie-Mellon University (1989)

[26] Quinn, C.J., Kiyavash, N., Coleman, T.P.: Directed information graphs. IEEE Transactions on information theory **61**(12), 6887–6909 (2015)

[27] Ratliff, N., Bagnell, J.A., Srinivasa, S.S.: Imitation learning for locomotion and manipulation. In: 2007 7th IEEE-RAS International Conference on Humanoid Robots. pp. 392–397. IEEE (2007)

[28] Rhinehart, N., McAllister, R., Levine, S.: Deep imitative models for flexible inference, planning, and control. International Conference on Learning Representations (ICLR) (2020)

[29] Schreiber, T.: Measuring information transfer. Physical review letters **85**(2), 461 (2000)

[30] Shimizu, S., Hoyer, P.O., Hyvärinen, A., Kerminen, A., Jordan, M.: A linear non-gaussian acyclic model for causal discovery. Journal of Machine Learning Research **7**(10) (2006)

[31] Spirtes, P., Glymour, C.N., Scheines, R.: Causation, prediction, and search. MIT press (2000)

[32] Tank, A., Covert, I., Foti, N., Shojaie, A., Fox, E.: Neural granger causality for nonlinear time series. IEEE Transaction on Pattern Analysis and Machine Intelligence (2021)

[33] Zhang, J., Kumor, D., Bareinboim, E.: Causal imitation learning with unobserved confounders. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 12263–12274. Curran Associates, Inc. (2020)

[34] Ziebart, B.D., Maas, A.L., Bagnell, J.A., Dey, A.K.: Maximum entropy inverse reinforcement learning. In: Aaai. vol. 8, pp. 1433–1438. Chicago, IL, USA (2008)

A Appendix: Perception model

We trained a β-VAE [15] as the *Perception model* of our causal pipeline, which its architecture is shown in Table 4. As the environment provides high dimensional observations, after the pre-processing step explained in Section 5.4, we first resize each observation to 64×64 pixels. The model takes in this $64 \times 64 \times 1$ input tensor (it is a gray-scale image) and encodes it into vectors μ and σ, each with size 128. After sampling the latent vector Z, it is passed through the decoder layers to reconstruct the observation.

For CIM and CIM-MLP, β was set to 6. Adjusting larger value ($\beta > 6$) applies intense pressure on the latent bottleneck and limits the representation capacity, preventing us from analyzing the characteristics of causes of speed. We set $\beta = 1$ for CIM-entangled corresponding to the original VAE [18].

Table 4: Encoder and decoder network design of the *Perception model*

Encoder	Decoder
Input: $64 \times 64 \times 1$	Input: $\mathbb{R}^{128}$
4×4 conv, 32 LReLU, stride 2, padding 1	128×256 FC, LReLU
4×4 conv, 32 LReLU, stride 2, padding 1	256×256 FC, LReLU
4×4 conv, 32 LReLU, stride 2, padding 1	$256 \times 64 * 4 * 4$ FC, LReLU
4×4 conv, 64 LReLU, stride 2, padding 1	4×4 deconv, 32 LReLU, stride 2, padding 1
$64 * 4 * 4 \times 256$ FC, LReLU	4×4 deconv, 32 LReLU, stride 2, padding 1
256×256 FC, LReLU	4×4 deconv, 32 LReLU, stride 2, padding 1
$256 \times 2 * 128$ FC	4×4 deconv, 1 LReLU, stride 2, padding 1

B Appendix: Speed predictor

B.1 Reducing regression problem to classification problem

In our experiments, we observed that the expert car's speed is either around 5.3 $\frac{m}{s}$ or almost 0. Hence, in order to avoid adding unnecessary complexity to the prediction task, we convert the target variable (speed) from continuous to categorical. As a result, the prediction task is turned into a classification problem. This modification requires some changes in *Controller*, as explained in Appendix C.

B.2 Model architecture

We design two different architectures to classify the next speed; each has an encoding layer and a classifier. A fully-connected encoder for each input variable in the encoding layer takes the concatenated last three steps of that variable and encodes it to the hidden representation. For CIM and CIM-entangled, each encoder is a 3×16 linear model. We also set every encoder as a 3×8

linear model for CIM-MLP. All of them have Leaky ReLU non-linearity. We separated encoders in CIM-MLP because we observed that the achieved test accuracy is increased in this manner.

The classifier is a fully-connected network fed by the concatenated hidden representations and classifies the speed of the next step. The details of classifier networks are presented in Table 5.

Table 5: Detailed architecture of classifiers

In CIM and CIM-entangled	In CIM-MLP
Input: $\mathbb{R}^{32}$	Input: $\mathbb{R}^{1024}$
32×16 FC, ReLU	1024×256 FC, ReLU
16×16 FC, ReLU	256×256 FC, ReLU
16×1 FC, Sigmoid	256×1 FC, Sigmoid

C Appendix: Reforming controller

As we stated in Section B.1, the speed prediction task is turned into a classification, and $\hat{S}^t_{\text{causal}}$ becomes a binary variable. Therefore, we need to reformulate Eq. 2 by replacing $\hat{S}^t_{\text{causal}}$ with $S_* \times \hat{S}^t_{\text{causal}}$ where $S_* = 5.3$:

$$\hat{S}^t_{\text{forward}} \leftarrow (1-\alpha)\hat{S}^t_{\text{trajectory}} + \alpha S_* \hat{S}^t_{\text{causal}}. \tag{4}$$

Note when $\hat{S}^t_{\text{causal}} = 1$, the above formula becomes $\hat{S}^t_{\text{forward}} \leftarrow \hat{S}_{\text{trajectory}} + \alpha\big[S_* - \hat{S}_{\text{trajectory}}\big]$, and when $\hat{S}^t_{\text{causal}} = 0$, then $\hat{S}^t_{\text{forward}} \leftarrow (1-\alpha)\hat{S}_{\text{trajectory}}$. However, in this formulation, a single wrong prediction can directly affect the overall performance. Thus, in order to alleviate the impact of wrong predictions, we revised the above formula by replacing $\hat{S}^t_{\text{causal}}$ with a smoothed version, $\tilde{S}^t$,

$$\tilde{S}^t = (1-\gamma)\hat{S}^t_{\text{causal}} + \gamma\tilde{S}^{t-1}, \tag{5}$$

where γ is a constant smoothing factor between 0 and 1, and $\tilde{S}^t$ is the exponential moving average of speed predictions.

D Appendix: Baselines and expert demonstrator

We built our work on *OATomobile* [9], a framework for autonomous driving research. To collect the demonstrations, we executed a rule-based autopilot expert implemented by [9] in `Town01`. DIM [28] and RIP [9] agents used in the experiments are designed in OATomobile, and we trained them on the collected dataset.

To all reviewers: We would like to thank you for taking the time to review the paper, and for the helpful comments. We believe that addressing the comments has resulted in improving the clarity and presentation of the paper's contributions. In the following, we will elaborate on our new experiment's result.

As suggested by one of the reviewers, we executed CIM, CIM-MLP, and CIM-entangled on `Town01` and `Town02` by setting α to 1, i.e., the ego car's speed equals the $\hat{S}_{\text{causal}}$. The new results show significant improvement of CIM and CIM-MLP.

As shown in the table, CIM has better performance than CIM-MLP in both towns, offering the advantage of incorporating causal relationships in training ML models.

Moreover, CIM-entangled's performance notably gets worse, which indicates that the speed predictor's proficiency is weakened. This is because each latent dimension captures both nuisance variables and causes; consequently, the selected parents set carries little information to predict the speed.