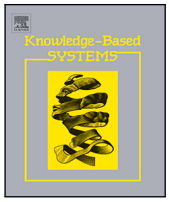
Knowledge-Based Systems 303 (2024) 112406

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



# Causal deconfounding deep reinforcement learning for mobile robot motion planning

Wenbing Tang [a,b], Fenghua Wu [b], Shang-wei Lin [b], Zuohua Ding [c], Jing Liu [a,*], Yang Liu [b], Jifeng He [a]

[a] Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, 200062, China
[b] College of Computing and Data Science, Nanyang Technological University, Singapore, 639798, Singapore
[c] School of Computer Science and Technology, Zhejiang Sci-Tech University, Hangzhou, 310018, China

ARTICLE INFO

*Keywords:*
Backdoor paths
Causal inference
Deep reinforcement learning
Mobile robots
Motion planning

ABSTRACT

Deep reinforcement learning (DRL) has emerged as an efficient approach for motion planning in mobile robot systems. It leverages the offline training process to enhance real-time computation efficiency. In DRL-based methods, the DRL models are trained to compute an action based on the current state of the robot and the surrounding obstacles. However, the trained models may capture spurious correlations through potential confounders, resulting in non-robust state representations, which limits the models' robustness and generalizability. In this paper, we propose a Causal Deconfounding DRL method for Motion Planning, CD-DRL-MP, to address spurious correlations and learn robust and generalizable policies. Specifically, we formalize the temporal causal relationships between states and actions using a structural causal model. We then extract the minimal sufficient state representation set by blocking the backdoor paths in the causal model. Finally, using the representation set, CD-DRL-MP learns the causal effect between states and actions while mitigating the detrimental influence of potential confounders and computes motion commands for mobile robots. Comprehensive experiments show that the proposed method significantly outperforms non-causal DRL methods and existing causal methods, while guaranteeing good robustness and generalizability.

## 1. Introduction

Motion planning is essential for mobile robots to complete tasks [1,2]. It aims to control robots to move from their initial positions to the given targets without causing collisions [1,3]. Many motion planning methods have been proposed in the literature. These methods can be classified into two main categories: conventional and learning-based methods. Conventional methods generally rely on the models of robots and/or the environment. Because they usually incur high computational costs and cannot guarantee real-time efficiency, their application to crowded and dynamic environments is inefficient. Instead, by combining reinforcement learning with deep neural networks, deep reinforcement learning (DRL) achieves a better trade-off between motion effectiveness and computation efficiency and becomes a promising technology for robot motion planning [4–6].

In general, DRL-based motion-planning methods aim to learn policy mapping from the state space (i.e., the states of the robot and obstacles) to the action space [3,7]. At any instant in time, DRL methods can infer an action based on the current composite state of the robot and environmental obstacles without considering the historical information of the planning process. However, such information can act as a potential confounder [8] and may result in DRL models learning easy-to-fit spurious correlations between states and actions [9] if the confounders are ignored, which is a problem largely ignored by existing methods.

Unfortunately, spurious correlations learned by ignoring potential confounders will cause robustness and generalization problems for the trained motion policy. On the one hand, the spurious correlations could induce the model to learn non-robust state representations, including irrelevant noise features that have no causal effect on action generation, but may interfere with decision-making [10,11]. Moreover, if we cannot formalize the state representations appropriately, the noise variables will affect the generated actions, resulting in unpredictable behavior. Owing to various disturbances and sensor attacks in current adversarial environments [12,13], real-time and robust motion planning is necessary. On the other hand, with spurious correlations, the trained DRL model cannot be generalized to other environments

* Corresponding author.
*E-mail addresses:* wbtang@stu.ecnu.edu.cn (W. Tang), fenghua.wu@ntu.edu.sg (F. Wu), shang-wei.lin@ntu.edu.sg (S.-w. Lin), zouhuading@hotmail.com (Z. Ding), jliu@sei.ecnu.edu.cn (J. Liu), yangliu@ntu.edu.sg (Y. Liu), jifeng@sei.ecnu.edu.cn (J. He).

https://doi.org/10.1016/j.knosys.2024.112406
Received 3 March 2024; Received in revised form 17 August 2024; Accepted 18 August 2024
Available online 22 August 2024

(i.e., environments with a different number of obstacles than the training ones). In real-world applications, since mobile robots are generally deployed in dynamic and changing environments, it is critical to guarantee consistently good performance in a variety of dynamic environments [14]. Hence, potential confounders must be considered in DRL-based motion planning methods.

In response to these challenges in the existing DRL methods, we focus on mitigating the potential confounders and propose the first Causal Deconfounding DRL method for Motion Planning, CD-DRL-MP. Specifically, we first build a structural causal model to formalize the temporal causal relationships among the factors involved in a motion planning task. Hence, we can find the potential confounders in the historical information and understand their effect on the generation of motion actions via the backdoor paths. Second, we develop a procedure to extract the representation of causal states that block all backdoor paths at each time step. Finally, using the causal state representation, CD-DRL-MP learns a DRL policy to select an action that maximizes the value function. CD-DRL-MP consists of two modules: (1) the deconfounding module, which extracts the expected causal information while mitigating the detrimental effect of spurious correlations, and (2) the planning module, which applies different motion planning algorithms for generating collision-free actions.

We conduct comprehensive experiments in various scenarios to evaluate the effectiveness, robustness, generalizability, and compatibility of CD-DRL-MP. To evaluate the efficiency of our method, we also compare it with other existing causal methods. The results show that (1) CD-DRL-MP outperforms the non-causal model and significantly boosts the motion-planning performance, improving the success rate from 89.226% to 95.313% and reducing the failure rate from 10.774% to 4.687%; (2) CD-DRL-MP can enhance the DRL models' robustness and maintain a relatively stable performance against disturbances; (3) CD-DRL-MP can generalize well beyond the training environment and achieve smaller performance degradation than the non-causal model (0.751% vs 5.099%); (4) the proposed deconfounding module is compatible with different DRL-based motion planning methods; (5) the proposed causal state representation is the minimal sufficient deconfounding set to learn the causal effect between states and actions; (6) the proposed method works well with real-world robot models, and (7) the proposed method outperforms existing causal methods.

The contributions of this paper are listed as follows:

- Based on analyzing the temporal causal relationships among the factors involved in a motion planning task, we formalize the motion planning problem via a causal lens to discover the presence of potential confounders and explain why a non-causal DRL method fails to learn the expected causal effect;
- We characterize a minimal sufficient deconfounding set of state representations for policy learning by blocking backdoor paths between states and actions at each time step;
- We propose a causal DRL method for motion planning, CD-DRL-MP, to learn a robust and generalizable DRL policy. The novelty of our method is that it can precisely capture the pure causal effect between states and actions while mitigating the detrimental effect of potential confounders.

The remainder of this paper is organized as follows. Section 2 summarizes the related work. Section 3 presents the theoretical basis and problem statement. Sections 4 and 5 present the detailed procedures for causal modeling and causal deconfounding, respectively. The experimental results are described in Section 6. The conclusion and future work are finally provided in Section 7.

## 2. Related work

**DRL motion planning.** Motion planning is an important problem in robotics, and several methods have been proposed by researchers. Some focus on generating feasible actions by considering local collision avoidance [15,16], whereas others consider optimal motion actions that require high computational costs [17,18]. Recently, because of the better trade-off between smooth motion and computational efficiency [7,19], DRL has been applied to motion planning. According to their inputs, the current DRL methods can be divided into two categories: sensor-based DRL methods [20,21], which use raw data from sensors as inputs, and agent-based DRL methods [4,6,7,19], which take the preprocessed agent-level information (e.g., types and states) as inputs. In practice, with such agent-level information, the DRL models can make more precise decisions [22]. However, the challenge in agent-based DRL models is the varying number of obstacles. To handle a varying number of obstacles, the authors of [6] proposed a DRL method with a Long Short-Term Memory (LSTM) model, in which the LSTM model was used to transform the obstacle states into a fixed-size hidden state based on their distances to the robot. Xu et al. [7] proposed a novel LSTM model to encode a varying number of obstacles, where the states of obstacles are ordered according to their collision risks with the robot. In [19], an attention-based method that transfers the input states of obstacles to a fixed-size embedding vector was proposed. However, these methods may capture spurious correlations between states and actions because of the presence of potential confounders [8].

**Causality for DRL.** DRL is a powerful machine learning methodology that has witnessed significant progress in the artificial intelligence community [23–26]. Its main idea is to learn a policy mapping states to actions while maximizing a cumulative reward. Recently, the combination of causality and DRL has been investigated. A major challenge when deploying causality-based DRL in real-world scenarios is the presence of confounders [27,28]. In strict-batch settings, this problem has been studied under the umbrella of causal imitation learning. Several methods have been proposed for introducing causality into behavior cloning [29,30], inverse reinforcement learning [31], and adversarial imitation learning [8]. For multi-agent cooperative settings, the authors in [32] proposed a backdoor adjustment-based deconfounded training method for credit assignment. Another challenge is generalizing DRL policies to unseen environments. To address this challenge, Lu et al. [9] proposed to build invariant predictors in the framework of invariant causal representation learning, and Bica et al. [33] proposed a method to learn a shared invariant representation of causal state features across various training environments. However, different from these works, we propose the first causal DRL method for motion planning to mitigate the detrimental effect of potential confounders introduced by temporal causal relationships.

## 3. Preliminaries and problem statement

### 3.1. Preliminaries on motion planning

**Motion planning.** We consider the motion planning problem for a holonomic mobile robot moving from its initial position $\mathbf{p}_i = [x_i, y_i]$ towards the target position $\mathbf{p}_g = [x_g, y_g]$ in an unknown environment with obstacles (e.g., pedestrians and other robots). We assume that the safety radius of the robot is $r$. The continuous motion time can be divided into a set of discrete time instants with an equal time step $\Delta t$. At each time instant $t$, $t \in \{0, 1, 2, \dots\}$, the robot computes a collision-free velocity for the duration $[t\Delta t, (t+1)\Delta t)$.

**State and action spaces.** The state and action of the robot at time instant $t$ can be represented as $\mathbf{s}_t = [\mathbf{p}_t, \mathbf{v}_t^-, \mathbf{p}_g, v_f, r] \in \mathbb{R}^8$ and $\mathbf{v}_t$, respectively, where $\mathbf{p}_t = [x_t, y_t]$ denotes the robot's position at $t$, $\mathbf{v}_t^-$, which satisfies $\mathbf{v}_t^- = \mathbf{v}_{t-1} = [vx_{t-1}, vy_{t-1}]$ for $t \geq 1$ and $\mathbf{v}_0^- = [0, 0]$, is the velocity of the robot in duration $[(t-1)\Delta t, t\Delta t)$, $v_f$ denotes the preferred speed, and $\mathbf{v}_t$ is the velocity that should be determined at $t$. Suppose that the obstacles detected at $t$ are $\{o_t^1, \dots, o_t^n\}$, and their states are denoted as $\mathbf{S}_t = \{\mathbf{so}_t^1, \dots, \mathbf{so}_t^n\}$, where $\mathbf{so}_t^j = [\mathbf{p}_t^j, \mathbf{v}_t^j, r^j] \in \mathbb{R}^5$; $\mathbf{p}_t^j = [x_t^j, y_t^j]$, $\mathbf{v}_t^j = [vx_t^j, vy_t^j]$, and $r^j$ are the position, velocity and radius of $o_t^j$, respectively. All possible values of composite state $(\mathbf{s}_t, \mathbf{S}_t)$ form the state space $\mathbb{S}^\pi$, and all candidate actions form the action space $\mathbb{A}$.

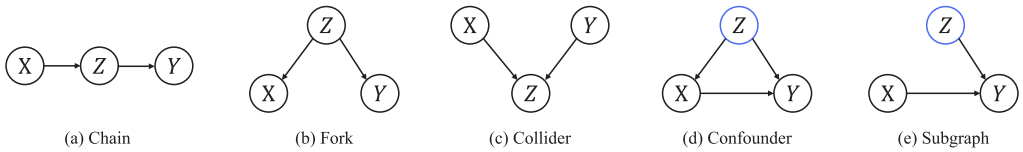



**Fig. 1.** Causal graphs: (a)–(c) The three basic structures of causal graphs; (d) Causal graph with a confounder (node $Z$); (e) Subgraph by removing the incoming edges of $X$.

**Optimization problem of motion planning.** Hence, the motion planning problem involves determining a motion policy $\pi : \mathbb{S}^{\pi} \mapsto \mathbb{A}$, $\forall \mathbf{S}_t^{\pi} \in \mathbb{S}^{\pi}$, $\pi(\mathbf{S}_t^{\pi}) = \mathbf{v}_t \in \mathbb{A}$, mapping each state (i.e., $\mathbf{S}_t^{\pi} = \mathbf{s}_t \oplus \mathbf{S}_t$, where $\oplus$ is the vector concatenation operation) to a collision-free action, such that the robot can arrive at the target as soon as possible without collisions. The set of all policies that take actions in $\mathbb{A}$ given states in $\mathbb{S}^{\pi}$ form a policy space $\Pi$. Formally, the motion planning task can be described as:

$$\arg\min_{\pi \in \Pi} T \tag{1}$$

$$s.t. \quad \mathbf{p}_0 = \mathbf{p}_i, \ \mathbf{p}_T = \mathbf{p}_g; \tag{2}$$

$$\|\mathbf{p}_t - \mathbf{p}_t^j\| \geq r + r^j, j \in \{0, 1, \ldots\}, \forall t\{1, \ldots, T\}; \tag{3}$$

$$\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{v}_t \Delta t, \forall t \in \{0, 1, \ldots, T-1\}. \tag{4}$$

where motion time $T$ in (1) is the optimization objective, (2) is the position constraints, (3) is the safety (collision avoidance) constraints, and (4) is the kinematics of the robot.

**DRL for motion planning.** According to [6,7,19], the optimization problem in (1)–(4) can be solved efficiently in the DRL framework as follows:

$$\mathbf{v}_t^* = \arg\max_{\mathbf{v} \in \mathbb{A}} R(\mathbf{S}_t^{\pi}, \mathbf{v}) + \gamma^{\Delta t \cdot v_f} V^*(\mathbf{S}_{t+1,\mathbf{v}}^{\pi}) \tag{5}$$

where $R(\mathbf{S}_t^{\pi}, \mathbf{v})$ is the one-step reward obtained by performing an action $\mathbf{v}$ in the current state $\mathbf{S}_t^{\pi}$ (the corresponding next state is $\mathbf{S}_{t+1,\mathbf{v}}^{\pi}$), $\gamma \in [0, 1]$ is a discount factor, $V^*(\cdot)$ is the optimal value function. The reward function $R(\cdot)$ can be found in [7,19].

### 3.2. Preliminaries on causality

**Structural causal models.** A structural causal model $\mathbf{M}$ is a tuple $\langle \mathbf{U}, \mathbf{V}, \mathbf{F}, P(\mathbf{U}) \rangle$ [34], where $\mathbf{U}$ is a set of exogenous variables, $\mathbf{V}$ is a set of endogenous variables, and $\mathbf{F}$ is a set of structural functions. $\forall V_i \in \mathbf{V}$, $f_{V_i} \in \mathbf{F}$ computes the value of $V_i$ based on the values of its parents in $\mathbf{V}$ and $\mathbf{U}$, i.e., $V_i \leftarrow f_{V_i}(\mathbf{V}_{V_i}, \mathbf{U}_{V_i})$, where $\mathbf{V}_{V_i} \subseteq \mathbf{V}$ and $\mathbf{U}_{V_i} \subseteq \mathbf{U}$ are the parents of $V_i$ in the sets $\mathbf{V}$ and $\mathbf{U}$, respectively. The value of each variable $U_i \in \mathbf{U}$ is drawn from an exogenous distribution $P(U_i)$.

**Causal graphs and confounders.** Each structural causal model $\mathbf{M}$ is associated with a causal graph $G$, consisting of a set of nodes representing the variables in $\mathbf{U}$ and $\mathbf{V}$ and a set of directed edges between the nodes representing the functions in $\mathbf{F}$. Hence, a causal graph can be expressed visually using a directed acyclic graph. Figs. 1(a)–(c) show the three basic structures of the causal graph, i.e., chain, fork, and collider. More complex causal graphs can be constructed on the basis of these three basic structures. In a causal graph, if node $Z$ has outgoing edges for both $X$ and $Y$, we call $Z$ the *confounder* [34]. As shown in Fig. 1(d), $Z$ is a confounder of the causal relationship between $X$ and $Y$. A confounder will make the true causal effect $X \rightarrow Y$ mixed with the spurious correlation between $X$ and $Y$ induced by the fork $X \leftarrow Z \rightarrow Y$. We denote by $G[\overline{\mathbf{X}}]$ the subgraph obtained from $G$ by removing edges coming into nodes in $\mathbf{X}$, where $\mathbf{X} \subseteq (\mathbf{U} \cup \mathbf{V})$. $G[\underline{\mathbf{X}}]$ is the subgraph of $G$ by removing edges going out from nodes in $\mathbf{X}$. For instance, for the causal graph in Fig. 1(d), the subgraph with respect to node $X$, i.e., $G[\overline{X}]$, is shown in Fig. 1(e).

**$d$-separation criterion.** In a causal graph $G$, suppose $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are three subsets of the variable set $\mathbf{U} \cup \mathbf{V}$. $\mathbf{X}$ and $\mathbf{Y}$ are *$d$-separated* of $\mathbf{Z}$, if every path between $\mathbf{X}$ and $\mathbf{Y}$ is blocked by $\mathbf{Z}$ [34], denoted as $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_G$. As shown in Figs. 1(a)–(b), in the chain and fork, the path between $X$ and $Y$ is blocked if we condition on $Z$, which can be denoted as $X \perp\!\!\!\perp Y | Z$. As shown in Fig. 1(c), for a collider, conditioning on $Z$ introduces an association between $X$ and $Y$, i.e., $X \perp\!\!\!\perp Y$ and $X \not\perp\!\!\!\perp Y | Z$.

### 3.3. Problem statement

In DRL-based motion planning, historical information may generate potential confounders in decision-making, owing to the temporal causal relationships between states and actions. Formally, given a causal graph $G$ and policy space $\Pi$ for the motion planning problem, suppose $\mathbf{S}_t^{\pi} \rightarrow \mathbf{v}_t$ is a path in $G$. A subset $\mathbf{Z}$ of the node set of $G$ is said to satisfy the *backdoor* criterion, if and only if $\pi(\mathbf{v}_t | \mathbf{Z}) \in \Pi$ and $(\mathbf{S}_t^{\pi} \perp\!\!\!\perp \mathbf{v}_t | \mathbf{Z})_{G[\underline{\mathbf{S}_t^{\pi}}]}$. Clearly, the existence of backdoor paths means that there are confounders between $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$, and we cannot train a good DRL policy without addressing these confounders. A satisfied policy should remove the detrimental effect of backdoor paths and guarantee consistently good performance across different environments. Hence, our problem can be formulated as:

**Problem 1.** *Given a robot moving around in an unknown environment, design a causal DRL method to block the potential backdoor paths and learn a robust and generalizable control policy such that the robot can move towards the target position safely without any collision.*

In general, if all paths from $\mathbf{S}_t^{\pi}$ to $\mathbf{v}_t$ are blocked by $\mathbf{Z}$, then the causal policy can be learned to capture the pure causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$ via $\mathbf{Z}$. In the sequel, we first give the determination of $\mathbf{Z}$ via causal modeling for motion planning and then provide the details of our causal deconfounding DRL method for motion planning.

## 4. Causal modeling for motion planning

In this section, we consider motion planning from a causal perspective, using a general structural causal model to determine the intrinsic causal mechanisms between states and actions. We use a causal graph to formalize the temporal causal relationships in the motion planning task, as shown in Fig. 2.

In motion planning, at any time instant $t$, an action $\mathbf{v}_t$ is generated based on the current state $\mathbf{S}_t^{\pi}$. Hence, $\mathbf{S}_t^{\pi}$ is a parent of $\mathbf{v}_t$ in the causal graph. Furthermore, a previous action can provide useful information for the current action selection [27], e.g., the speed of a robot should not be increased when its speed reaches the maximum one at the previous time instant. Hence, we add a temporal causal relationship between $\mathbf{v}_{t-1}$ and $\mathbf{v}_t$. Additionally, because the action is limited by the physical constraints of the robot's actuators, we add an exogenous variable $U^{\mathrm{v}}$ that affects action generation. According to the Markov decision process of DRL [35], given the state $\mathbf{S}_t^{\pi}$ and the selected action $\mathbf{v}_t$, the DRL system will move to a new state $\mathbf{S}_{t+1}^{\pi}$ with a state transition probability $p(\mathbf{S}_{t+1}^{\pi} | \mathbf{S}_t^{\pi}, \mathbf{v}_t)$ and receive a reward $R_{t+1}$. Therefore, there are causal paths: $\mathbf{S}_{t-1}^{\pi} \rightarrow \mathbf{S}_t^{\pi}$, $\mathbf{v}_{t-1} \rightarrow \mathbf{S}_t^{\pi}$, $\mathbf{S}_t^{\pi} \rightarrow R_{t+1}$, and $\mathbf{v}_t \rightarrow R_{t+1}$.

Based on the above descriptions, we formalize the following structural causal model for the motion planning task:

$$\begin{cases} \mathbf{S}_t^{\pi} \leftarrow f(\mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}); \\ \mathbf{v}_t \leftarrow \pi(\mathbf{S}_t^{\pi}, \mathbf{v}_{t-1}; U^{\mathrm{v}}); \\ R_{t+1} \leftarrow r(\mathbf{S}_t^{\pi}, \mathbf{v}_t). \end{cases}$$

where $f(\cdot)$, $\pi(\cdot)$, and $r(\cdot)$ represent structural functions. Specifically, $f(\cdot)$ denotes the evolution of the system, which depicts the state transition

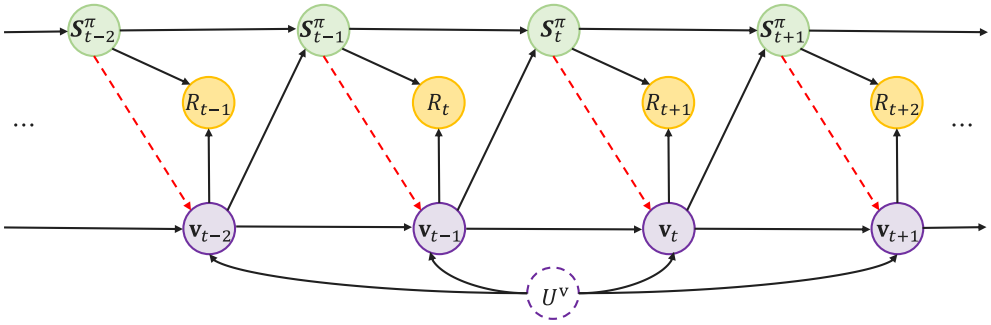



**Fig. 2.** Causal graph of the structural causal model illustrating the temporal causal relationships between states, actions, and rewards. The circles with solid and dashed lines represent the endogenous and exogenous variables, respectively. The causal effects of interest are colored in red.

of the DRL system from $t-1$ to $t$. Function $\pi(\cdot)$ is an action generation function that indicates the computation of an action. Given the current state $\mathbf{S}_t^{\pi}$ and action $\mathbf{v}_t$, the reward function $r(\cdot)$ is used to evaluate how good the action is at the current state. Note that for a DRL agent, functions $f(\cdot)$ and $r(\cdot)$ are unobserved. The DRL agent updates its policy network $\pi(\cdot)$ to maximize the cumulative reward. That is, the DRL network is expected to learn the causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$.

Unfortunately, following the causal graph in Fig. 2, the current DRL methods suffer from potential confounders and fail to learn the causal effect. As shown in Fig. 2, backdoor paths exist between $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$, e.g., $\mathbf{S}_t^{\pi} \leftarrow \mathbf{v}_{t-1} \rightarrow \mathbf{v}_t$ and $\mathbf{S}_t^{\pi} \leftarrow \mathbf{S}_{t-1}^{\pi} \rightarrow \mathbf{v}_{t-1} \rightarrow \mathbf{v}_t$. Let $\mathbf{H}_{t-1} = \{(\mathbf{S}_i^{\pi}, \mathbf{v}_i, R_{i+1}) | i = 0, \ldots, t-1\}$ be the historical information record of the DRL system at time instant $t$. Because $\mathbf{H}_{t-1}$ affects both $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$, i.e., $\mathbf{S}_t^{\pi} \leftarrow \mathbf{H}_{t-1} \rightarrow \mathbf{v}_t$, a non-causal model will capture spurious correlations by learning from the likelihood $P(\mathbf{v}|\mathbf{S}^{\pi})$. To observe this, according to the Bayes' rule:

$$P(\mathbf{v}|\mathbf{S}^{\pi}) = \sum_{h \in \mathbf{H}} P(\mathbf{v}|\mathbf{S}^{\pi}, h) P(h|\mathbf{S}^{\pi}). \tag{6}$$

where $\mathbf{H}_{t-1}$ introduces the bias via $P(h|\mathbf{S}^{\pi})$. Therefore, when potential confounders exist in $\mathbf{H}_{t-1}$, the non-causal DRL method learns easy-to-fit spurious correlations via backdoor paths. However, models based on spurious correlations may produce erroneous predictions [9,36,37]. In particular, they cannot be generalized to unseen environments [33,38]. To obtain a better policy, one should block the backdoor paths and train a causal policy to capture the pure causal effect.

## 5. Causal deconfounding DRL for motion planning

In this section, we propose a causal method, CD-DRL-MP, to learn a robust and generalizable DRL policy for motion planning. The key to training a causal policy is to block backdoor paths to estimate the causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$ at each timestep. Based on the structural causal model in Section 4, we make the following statements.

**Proposition 1.** *Given the current state $\mathbf{S}_t^{\pi}$, the current action $\mathbf{v}_t$ is not d-separated of the historical information $\mathbf{H}_{t-1}$, that is,*

$$P(\mathbf{v}_t|\mathbf{S}_t^{\pi}) \neq P(\mathbf{v}_t|\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}, R_t, \ldots, \mathbf{S}_0^{\pi}, \mathbf{v}_0, R_1). \tag{7}$$

**Proof.** Because the backdoor paths between $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$ are unblocked (e.g., $\mathbf{S}_t^{\pi} \leftarrow \mathbf{v}_{t-1} \rightarrow \mathbf{v}_t$), $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$ are dependent [34], i.e., not *d-separated*.

**Proposition 2.** *Given the set $\{\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}\}$, the current action $\mathbf{v}_t$ is independent of $\mathbf{H}_{t-1}$, i.e.,*

$$P(\mathbf{v}_t|\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}) = P(\mathbf{v}_t|\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}, R_t, \ldots, \mathbf{S}_0^{\pi}, \mathbf{v}_0, R_1). \tag{8}$$

**Proof.** 1 Considering two-time steps (e.g., $t$ and *t-1*), there are four backdoor paths, i.e., *Path1*: $\mathbf{S}_t^{\pi} \leftarrow \mathbf{v}_{t-1} \rightarrow \mathbf{v}_t$, *Path2*: $\mathbf{S}_t^{\pi} \leftarrow \mathbf{v}_{t-1} \leftarrow U^{\mathrm{v}} \rightarrow \mathbf{v}_t$, *Path3*: $\mathbf{S}_t^{\pi} \leftarrow \mathbf{S}_{t-1}^{\pi} \rightarrow \mathbf{v}_{t-1} \rightarrow \mathbf{v}_t$, and *Path4*: $\mathbf{S}_t^{\pi} \leftarrow \mathbf{S}_{t-1}^{\pi} \rightarrow \mathbf{v}_{t-1} \leftarrow U^{\mathrm{v}} \rightarrow \mathbf{v}_t$. According to the *d*-separation criterion [34], conditioning on $\mathbf{v}_{t-1}$ will block *Path1* as $\mathbf{v}_{t-1}$ is the middle node of a fork. For *Path2* and *Path3*, $\mathbf{v}_{t-1}$ is the middle node of chains. Therefore, the *Paths* 2 and 3 will also be blocked if we condition on $\mathbf{v}_{t-1}$. However, because $\mathbf{v}_{t-1}$ is a collider in *Path4*, conditioning on $\mathbf{v}_{t-1}$ will unblock this path. To block all backdoor paths, we further add $\mathbf{S}_{t-1}^{\pi}$ into the deconfounding set. Finally, conditioning on $\mathbf{v}_{t-1}$ and $\mathbf{S}_{t-1}^{\pi}$ will make *Paths 1-4* blocked. (2) Considering longer time steps, $\mathbf{v}_{t-1}$ and $\mathbf{S}_{t-1}^{\pi}$ can also block all backdoor paths. Hence, given the set $\{\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}\}$, $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$ are *d-separated*. This implies that $\mathbf{v}_t$ is independent of $\mathbf{H}_{t-1}$.

Let $\mathbf{S}_{t,causal}^{\pi}$ denote the set $\{\mathbf{S}_t^{\pi}, \mathbf{S}_{t-1}^{\pi}, \mathbf{v}_{t-1}\}$. According to Propositions 1 and 2, $\mathbf{S}_{t,causal}^{\pi}$ blocks all backdoor paths between $\mathbf{S}_t^{\pi}$ and $\mathbf{v}_t$, i.e., $(\mathbf{S}_t^{\pi} \perp\!\!\!\perp \mathbf{v}_t|\mathbf{S}_{t,causal}^{\pi})_{G[\underline{\mathbf{S}_t^{\pi}}]}$.

**Proposition 3.** *$\mathbf{S}_{t,causal}^{\pi}$ is the minimal sufficient deconfounding set to learn the causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$.*

**Proof.** First, as discussed in Proposition 1, $\mathbf{S}_{t,causal}^{\pi}$ contains all causal information of $\mathbf{H}_{t-1}$. Hence, it can be used to estimate the causal effect. Second, by deleting any element from $\mathbf{S}_{t,causal}^{\pi}$ will result in a backdoor path being unlocked. For example, deleting $\mathbf{S}_{t-1}^{\pi}$ will cause *Path 4* being unblocked. Removing $\mathbf{v}_{t-1}$ will open backdoor paths through *Paths 1-3*. Additionally, $\mathbf{S}_t^{\pi}$ is necessary to learn the causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$. Hence, $\mathbf{S}_{t,causal}^{\pi}$ is the minimal sufficient deconfounding set.

According to Proposition 3, we can use $\mathbf{S}_{t,causal}^{\pi}$ to train a causal DRL policy $\pi(\mathbf{v}_t|\mathbf{S}_{t,causal}^{\pi})$. It can capture the pure causal effect of $\mathbf{S}_t^{\pi}$ on $\mathbf{v}_t$, which is fundamentally different from current DRL methods based on the posterior probability $P(\mathbf{v}|\mathbf{S}^{\pi})$. Based on the minimal sufficient deconfounding set $\mathbf{S}_{t,causal}^{\pi}$, we propose our causal deconfounding DRL motion planning method CD-DRL-MP. Note that CD-DRL-MP can be instantiated with any DRL-based motion planning method. In the sequel, we take the DQN-based motion planning method LSTM-RL [6], as an example to demonstrate the training and inference processes of CD-DRL-MP. More experiments instantiating CD-DRL-MP with different existing DRL-based motion planning methods are provided in Section 6.5.

Specifically, at any time instant $t$, $\mathbf{S}_t$ and $\mathbf{S}_{t-1}$ are first sent to an LSTM model and transformed into fixed-size hidden states; then, $\mathbf{S}_{t,causal}^{\pi}$ can be obtained by concatenating the resulting hidden states, the robot's states ($\mathbf{s}_t$ and $\mathbf{s}_{t-1}$), and the previous action $\mathbf{v}_{t-1}$; third, $\mathbf{S}_{t,causal}^{\pi}$ is sent to the DRL model to select the new action $\mathbf{v}_t$. The framework of CD-DRL-MP is shown in Fig. 3.

The training process of CD-DRL-MP is described in Algorithm 1. It uses the DQN (Deep Q-Network) algorithm to train a Q-network that approximates the action-value function. The memory replay and target network techniques are applied to enhance learning efficiency and improve learning stability. Specifically, the replay memory $E$, Q-network $Q$, and target network $\hat{Q}$ are initialized first (lines 1 and 2),

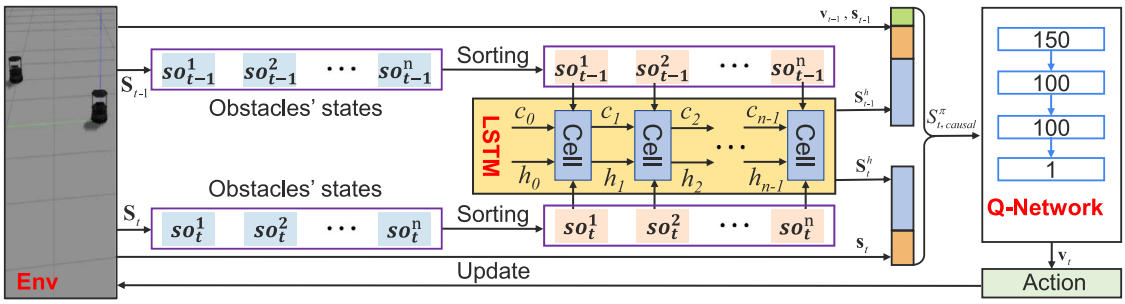



**Fig. 3.** Framework of the proposed causal deconfounding DRL motion planning method CD-DRL-MP.

**Algorithm 1** Training process of CD-DRL-MP

**Input:** Training episodes $N_e$, replay iterations $N_r$, the action space $\mathbb{A}$, the exploration probability $\epsilon$, the maximal time steps $T_m$, and the update frequency for the target network $\omega$;
**Output:** Trained motion planning method CD-DRL-MP;
1: Collect a set of motion trajectories $L$ via ORCA;
2: Initialize a replay memory $E$, a Q-network $Q$, and a target network $\hat{Q}$ on $L$;
3: **for** $episode = 1 : N_e$ **do**
4: **for** $ite = 1 : N_r$ **do**
5: Sample $\mathbf{s}_0$, $\mathbf{S}_0$, and $\mathbf{p}_g$;
6: $done = False$, $t = 0$, $Traj = \{(\mathbf{s}_0, \mathbf{S}_0)\}$;
7: Generate a random action $\mathbf{v}_0$ and move to $(\mathbf{s}_1, \mathbf{S}_1)$;
8: **while** *not done* **do**
9: Retrieve states $\mathbf{s}_t$ and $\mathbf{S}_t$;
10: $\mathbf{S}_t^h, \mathbf{S}_{t-1}^h \leftarrow LSTM(\mathbf{S}_t), LSTM(\mathbf{S}_{t-1})$;
11: Generate the state representation $\mathbf{S}_{t,causal}^{\pi}$ ;
12: Generate $\mathbf{v}_t$ using $\hat{Q}$ and the $\epsilon$-greedy policy;
13: The robot moves to $\mathbf{s}_{t+1}$ controlled by $\mathbf{v}_t$;
14: $t \leftarrow t+1$, $Traj \leftarrow Traj \cup \{(\mathbf{s}_{t+1}, \mathbf{S}_{t+1})\}$;
15: **if** *collided* or *reached* or $t > T_m$ **then**
16: $done = True$;
17: **end if**
18: **end while**
19: **if** *collided* or *reached* **then**
20: Update the memory $E$ with $Traj$;
21: **end if**
22: **end for**
23: Update $Q$ and $LSTM$ by gradient descent with $E$;
24: **if** $(episode \bmod \omega) == 0$ **then**
25: Update target network $\hat{Q} \leftarrow Q$;
26: **end if**
27: **end for**

**Algorithm 2** Motion planning with CD-DRL-MP

**Input:** The well-trained LSTM model and the DRL model based on $Q$, the action space $\mathbb{A}$, the exploration probability $\epsilon$, the preferred speed $v_f$, the initial and target positions $\mathbf{p}_0$ and $\mathbf{p}_g$;
**Output:** The generated trajectory $\Gamma$;
1: Retrieve the surrounding obstacles $\mathbf{S}_0$
2: done = False; t = 0; $\mathbf{s}_0 = [\mathbf{p}_0, \mathbf{0}, \mathbf{p}_g, v_f, r]$;
3: $\Gamma \leftarrow \Gamma \cup \{\mathbf{s}_0\}$;
4: Generate a random action $\mathbf{v}_0$ and move to $(\mathbf{s}_1, \mathbf{S}_1)$; $t \leftarrow t+1$;
5: **while** *not done* **do**
6: Sort $\mathbf{S}_t$ and $\mathbf{S}_{t-1}$ using the LSTM model;
7: Generate the state representation $\mathbf{S}_{t,causal}^{\pi}$ ;
8: Select $\mathbf{v}_t$ from $\mathbb{A}$ via the DRL model trained in Algorithm 1 ;
9: $\mathbf{s}_t = [\mathbf{p}_t, \mathbf{v}_t, \mathbf{p}_g, v_f, r]$; $\Gamma \leftarrow \Gamma \cup \{\mathbf{s}_t\}$;
10: Move to next position $\mathbf{p}_{t+1}$;
11: $t \leftarrow t+1$;
12: **if** $||\mathbf{p}_{t-1} - \mathbf{p}_g|| < \varepsilon$ or *collided* **then**
13: done = True;
14: **end if**
15: **end while**
16: **return** $\Gamma$.

using a set of trajectories generated from Optimal Reciprocal Collision Avoidance (ORCA) [18]. Second, the algorithm repeats $N_e$ episodes to train the DRL model. We replay the robot motion $N_r$ times for each episode and then update the memory $E$. In each replay, we reset the initial state $\mathbf{S}_0^{\pi}$ and the target position $\mathbf{p}_g$ randomly, and then control the robot to move to $\mathbf{p}_g$ (lines 5–18). At any time instant $t$, the algorithm computes the hidden states $\mathbf{S}_t^h$ and $\mathbf{S}_{t-1}^h$ via the LSTM model given in [6] (line 10); then, the algorithm obtains the causal state representation $\mathbf{S}_{t,causal}^{\pi}$ (line 11). The $\epsilon$-greedy policy was used to select $\mathbf{v}_t$ (line 12), which can be described as follows:

$$\mathbf{v}_t = \begin{cases} \text{randomly select from } \mathbb{A}, & \text{with probability } \epsilon; \\ \arg\max_{\mathbf{v}\in\mathbb{A}} Q(\mathbf{S}_{t,causal}^{\pi}, \mathbf{v}), & \text{with probability } 1-\epsilon. \end{cases}$$

where $Q(\mathbf{S}_{t,causal}^{\pi}, \mathbf{v}) = R(\mathbf{S}_{t,causal}^{\pi}, \mathbf{v}) + \gamma^{\Delta t \cdot v_f} \cdot [\arg\max_{\mathbf{v}\in\mathbb{A}} \hat{Q}(\mathbf{S}_{t+1,\mathbf{v},causal}^{\pi}, v)]$. The robot moves with $\mathbf{v}_t$ for the duration $[t\Delta t, (t+1)\Delta t)$. When the robot collides with an obstacle, arrives at its target position, or reaches the maximal motion time, the current replay is terminated (lines 15–17). The generated trajectory is used to update replay memory $E$ (lines 19–21) only when the replay is terminated by the *collided* or *reached* status.

At the end of each episode, the weights of $Q$ and LSTM are updated simultaneously using the gradient descent strategy. The target network $\hat{Q}$ is updated at every $\omega$ episode (lines 24–26).

The motion planning process using CD-DRL-MP is given in Algorithm 2. At any time instant, the robot first perceives its state $\mathbf{s}_t$ and retrieves the state of the surrounding obstacles $\mathbf{S}_t$; Second, the current and previous states of the obstacles are first processed using a well-trained LSTM model (line 6). Third, the causal state representation $\mathbf{S}_{t,causal}^{\pi}$ is generated based on the robot states $\mathbf{s}_t$ and $\mathbf{s}_{t-1}$, the previous action $\mathbf{v}_{t-1}$, and the outputs of the LSTM model $\mathbf{S}_t^h$ and $\mathbf{S}_{t-1}^h$ (line 7). Subsequently, the Q-network trained in the DRL model takes the causal state representation as input, computes the corresponding values for all action candidates and returns the one with the maximal value (line 8). As time elapses, the robot moves forward with the new velocity in the next time duration, i.e., $\mathbf{p}_{t+1} = \mathbf{p}_t + \mathbf{v}_t \Delta_t$ (line 10). The planning process is completed when the robot reaches its target position within a given error tolerance threshold $\epsilon$ or a collision is detected during the movement from $\mathbf{p}_t$ to $\mathbf{p}_{t+1}$ (lines 12 and 13). Finally, the motion trajectory for the robot is generated (line 16).

## 6. Experimental evaluation

In this section, we conduct simulations to validate the performance of CD-DRL-MP under different scenarios. Specifically, we first evaluate the effectiveness of CD-DRL-MP with a varying number of obstacles (Section 6.2). Second, we demonstrate the robustness of CD-DRL-MP against eight disturbance attacks (Section 6.3). Third, by applying CD-DRL-MP to the seen and unseen environments, we evaluate its generalizability (Section 6.4). Fourth, we evaluate the compatibility of the proposed causal deconfounding module with two well-established

**Table 1**
Performance of CD-DRL-MP and the non-causal DRL model with different numbers of cases (non-causal/causal).

| # case | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|
| 100 | 0.860/**0.970** | 0.140/**0.030** | 0.000/0.000 | 8.108/**7.951** | 0.291/**0.361** |
| 200 | 0.875/**0.960** | 0.125/**0.040** | 0.000/0.000 | 8.080/**7.945** | 0.303/**0.355** |
| 300 | 0.893/**0.947** | 0.107/**0.053** | 0.000/0.000 | 8.090/**7.942** | 0.311/**0.344** |
| 400 | 0.890/**0.940** | 0.110/**0.060** | 0.000/0.000 | 8.086/**7.943** | 0.309/**0.341** |
| 500 | 0.890/**0.938** | 0.110/**0.062** | 0.000/0.000 | 8.089/**7.948** | 0.310/**0.342** |
| 600 | 0.890/**0.940** | 0.110/**0.060** | 0.000/0.000 | 8.080/**7.949** | 0.309/**0.343** |
| 700 | 0.891/**0.940** | 0.109/**0.060** | 0.000/0.000 | 8.077/**7.948** | 0.310/**0.343** |
| 800 | 0.891/**0.940** | 0.109/**0.060** | 0.000/0.000 | 8.075/**7.958** | 0.311/**0.343** |
| 900 | 0.896/**0.940** | 0.104/**0.060** | 0.000/0.000 | 8.072/**7.962** | 0.313/**0.344** |
| 1000 | 0.901/**0.941** | 0.099/**0.059** | 0.000/0.000 | 8.077/**7.962** | 0.317/**0.345** |
| 1100 | 0.898/**0.983** | 0.102/**0.017** | 0.000/0.000 | 8.087/**8.033** | 0.315/**0.392** |
| 1200 | 0.901/**0.982** | 0.099/**0.018** | 0.000/0.000 | 8.085/**8.034** | 0.317/**0.392** |
| 1300 | 0.902/**0.982** | 0.098/**0.018** | 0.000/0.000 | 8.083/**8.034** | 0.317/**0.392** |
| 1400 | 0.902/**0.946** | 0.098/**0.054** | 0.000/0.000 | 8.083/**7.965** | 0.317/**0.348** |
| 1500 | 0.904/**0.948** | 0.096/**0.052** | 0.000/0.000 | 8.084/**7.966** | 0.319/**0.349** |
| Average | 0.892/**0.953** | 0.108/**0.047** | 0.000/0.000 | 8.083/**7.969** | 0.311/**0.356** |

DRL methods under two simulation scenarios (Section 6.5). Then, we conduct six ablation studies to validate the design of the proposed causal method (Section 6.6). In Section 6.7, we evaluate the proposed method through realistic simulation experiments using a Gazebo simulator with Robot Operating System (ROS). Finally, we compare our method with two other causal methods (Section 6.8).

### 6.1. Simulation setup

We implement CD-DRL-MP in the simulation environment CrowdNav [19], a well-established robot simulator that enables the training and benchmarking of various DRL-based motion planning algorithms. We run all simulations on an Ubuntu 18.04.6 LTS system with an NVIDIA RTX A4000 GPU, an Intel(R) Core(TM) i9-13900K CPU with a 5.80 GHz processor, and 32 GB of memory. All models are implemented in PyTorch and use the SGD optimizer to update the network weights. For the configuration of the robots, we use the default values from the simulated environment, with a preferred speed of $v_f = 1$ and a safe radius of $r = 0.3$. The candidate action space is $\mathbb{A} = \{(v_i \cos\theta_j, v_i \sin\theta_j) | v_i = (e^{i/5} - 1)/(e - 1), \theta_j = j/8, i = 1, \ldots, 5, j = 0, \ldots, 15\}$. For training the DRL model, $N_e = 2{,}000$, $N_r = 1$, $\omega = 50$, $T_m = 100$, $\gamma = 0.9$, the dimension of the LSTM's hidden state is 50, $c_0$ and $h_0$ are initialized with zero vectors, and the architectures of $Q$ and $\hat{Q}$ are both $(150, 100, 100, 1)$. During the training stage, the LSTM and DRL models are trained simultaneously with a learning rate of 0.001.

### 6.2. Effectiveness of CD-DRL-MP

To evaluate the effectiveness of CD-DRL-MP, we compare CD-DRL-MP with the corresponding non-causal DRL method in different environments. Note that the non-causal DRL (LSTM-RL [6]) is trained with the same hyper-parameters in the training stage of CD-DRL-MP. Specifically, we train the causal and non-causal models with a varying number of obstacles, ranging from 1 to 10. The metrics to evaluate the experiments are as follows. The *success* rate is used to quantify the proportion of successful runs (i.e., reaching the target position safely within the given maximal motion time) among all testing runs. The *collision* rate quantifies the proportion of collided runs among all runs. The *timeout* rate measures the proportion of runs that exceed the time limit of all the runs. The *navTime* metric is defined as the average navigation time of successful runs, and represents the time consumed by the robot to travel from the initial position to the target position. The cumulative *reward* is a measure of how well each action is generated by DRL throughout the planning process.

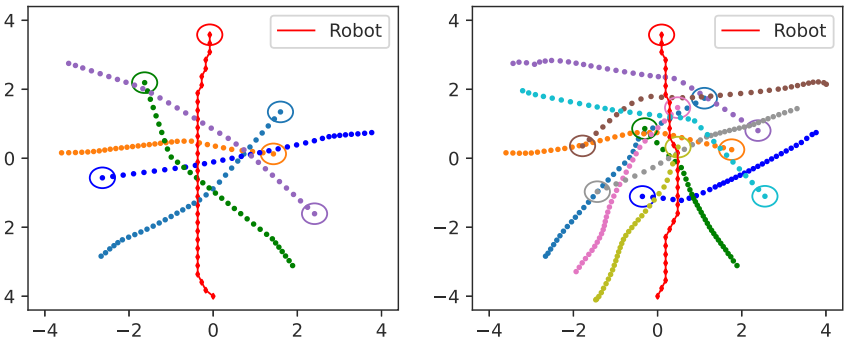
Table 1 lists the testing results of both models on the 15 test sets, where the number of test cases varied from 100 to 1500. A test case is an initial configuration of a simulation scenario, which determines the initial and target positions, and initial velocity of each obstacle. Hence, a test case with $n$ obstacles can be noted as $test\_case = \{\mathbf{p}_i^1, \mathbf{v}_0^1, \mathbf{p}_g^1; \cdots; \mathbf{p}_i^n, \mathbf{v}_0^n, \mathbf{p}_g^n\}$. A group of test cases forms a test set, i.e., $TEST\_SET = \{test\_case_1, \ldots, test\_case_m\}$. In each test set, the number of obstacles varies from 1 to 10. In our simulations, all obstacles are randomly located on the circumference of a circle and need to move to the opposite side of the circle (i.e., circle-crossing [19]). From the table, we make the following observations:

- **CD-DRL-MP can improve motion planning performance significantly.** The table shows that CD-DRL-MP can obtain higher success rates on all test sets and improve the success rate from 89.226% to 95.313% on average. Moreover, CD-DRL-MP causes lower failure (i.e., collision + timeout) rates, i.e., 10.773% vs 4.687% on average, which means the robot usually falls into collisions or cannot arrive at its target position under the control of the non-causal DRL model.
- **CD-DRL-MP can generate more efficient motion.** Among all the successful test sets, CD-DRL-MP has a lower navigation time than the non-causal model, saving 1.416% (8.083 s vs 7.969 s) on average. It means that the robot under the control of CD-DRL-MP can generate shorter trajectories to move to its target positions. Hence, CD-DRL-MP can generate more efficient motion for robots.
- **CD-DRL-MP can increase cumulative reward.** From the table, we can find that CD-DRL-MP can significantly increase the cumulative rewards, improving the average reward from 0.311 to 0.356. It means that the robot under the control of CD-DRL-MP can (1) keep a safe distance from obstacles to ensure safety and (2) arrive at its destinations within the given time duration.

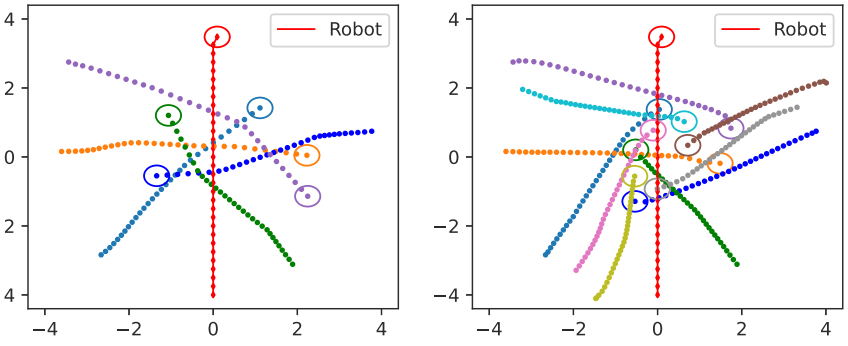
To further evaluate the statistical significance of the results, we conduct Levene's test for equal variances and T-test for equal means [39]. The results are listed in Table 2. The result is statistically significant, which means that there are significant differences in the evaluated metrics between CD-DRL-MP and the non-causal model. Hence, we can conclude that compared with the non-causal DRL method, CD-DRL-MP can achieve better performance and significantly improves the motion planning performance from different aspects.

In addition, we conduct ten independent test sets for the two methods, where the number of obstacles varied from 1 to 10. Each test set contains 500 test cases. The results are listed in Table 3. From the results, we can find that CD-DRL-MP can improve the motion planning performance in environments with different numbers of obstacles. Moreover, we found that with an increase in the number of obstacles, the non-causal model exhibited significant performance degradation, reducing the success rate from 99.4% to 89.6%, whereas CD-DRL-MP maintained high success rates (97.8% for one obstacle and 92.6% for ten obstacles).

In the following, we further show some trajectories computed by the two methods, as shown in Fig. 4. From Fig. 4(a), we can find that the




(a) Trajectories of the non-causal method.




(b) Trajectories of CD-DRL-MP.

**Fig. 4.** Trajectories generated by CD-DRL-MP and the non-causal method with five and ten obstacles, respectively. The red diamonds represent the robot's locations at different time instants. The initial and target positions of the robot are $[0, -4]$ and $[0, 4]$, respectively. The solid circles are the locations of the obstacles at each time instant. The robot is controlled by CD-DRL-MP, while all obstacles are controlled by ORCA. The hollow circles represent the final locations and safe radius of the robot and obstacles.

**Table 2**
Significance testing of each metric on 15 test sets.

| Metric | Method | Mean | Levene's Test | T-Test |
|---|---|---|---|---|
| Success | Non-causal | 0.892 | 0.027 | 6.492E−12 |
| | Causal | **0.953** | | |
| Failure | Non-causal | 0.108 | 0.027 | 6.492E−12 |
| | Causal | **0.047** | | |
| NavTime | Non-causal | 8.084 | 0.002 | 4.836E−13 |
| | Causal | **7.969** | | |
| Reward | Non-causal | 0.311 | 0.003 | 5.582E−09 |
| | Causal | **0.356** | | |

non-causal model makes the generated paths oscillating, which causes longer navigation time. In contrast, CD-DRL-MP can navigate the robot towards its target safely and smoothly, as shown in Fig. 4(b). Therefore, we can conclude that by blocking all backdoor paths, CD-DRL-MP can capture the causal effect between the states and actions and generate better control actions, resulting in smoother motion trajectories. Note that although there may be intersections on the trajectories of the robot and obstacles, such as the point $[0, -1]$ shown in the left part of Fig. 4(b), these intersections do not constitute a collision. This is because the robot and obstacles pass through these intersections at different time instants. For example, for the intersection point $[0, -1]$ in the left part of Fig. 4(b), both the robot (red path) and Obstacle 4 (green path) pass near this point. However, the robot passes the intersection at the 13th time instant, whereas Obstacle 4 passes it at the 20th instant. The dynamic display of the trajectories is available on the website https://sites.google.com/view/cd-drl-mp.

### 6.3. Robustness of CD-DRL-MP

To evaluate robustness, disturbance attacks were performed on the position of the robot and the performance of the CD-DRL-MP and non-causal model was compared. At any time instant $t$, the position received by the DRL model becomes:

$$\mathbf{p}'_t = \Delta(\mathbf{p}_t, i_t) = i_t \times D(\mathbf{p}_t) + (1 - i_t) \times \mathbf{p}_t \tag{9}$$

where $i_t \in \{0, 1\}$ is the label indicating whether the position is attacked at time $t$ or not, $D(\mathbf{p}_t) \in [\mathbf{p}_t - \delta, \mathbf{p}_t + \delta]$ is the attacked position, where $\delta$ is the attack radius. We assume that each $i_t$ follows a probability distribution $P^D$, indicating the attack level. For example, when $P^D$ is assigned 10%, position $\mathbf{p}_t$ has a 10% chance of being attacked. By setting concrete values for the attack level and attack radius, we can generate different disturbance scenarios to evaluate the robustness. Table 4 presents the performance of either model under an attack-free test set (i.e., $P^D = 0\%$ and $\delta = 0$) and eight attacked test sets. Each test set contains 500 test cases with a varying number of obstacles, from 1 to 10. From the table, we can find that CD-DRL-MP shows better performance on all test sets and maintains a relatively stable success rate. However, the performance of the non-causal model was significantly reduced when the strength of the disturbance attacks increased. This indicates that CD-DRL-MP can achieve more robust motion planning than the non-causal model, and can deal with regular disturbances in the data.

Fig. 5 shows the changes in success rate, failure rate, navigation time, and reward, with an increase in attack strength. We can observe that the non-causal model shows significant decreases in the success rate and reward, and significant increases in the collision rate and navigation time. This means that spurious correlations can lead to a model for learning non-robust policies, resulting in performance degradation. In contrast, CD-DRL-MP can maintain a relatively stable performance on all metrics when the strength of disturbance increases.

**Table 3**
Performance of CD−DRL−MP and the non-causal DRL model in ten test sets with different numbers of obstacles (non-causal/causal).

| Obstacles | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|
| 1 | **0.994**/0.978 | **0.006**/0.022 | 0.000/0.000 | **8.238**/8.298 | **0.399**/0.389 |
| 2 | 0.930/**0.932** | 0.070/**0.068** | 0.000/0.000 | 8.230/**8.011** | 0.337/**0.332** |
| 3 | 0.898/**0.904** | 0.102/**0.096** | 0.000/0.000 | 8.129/**7.915** | 0.306/**0.309** |
| 4 | 0.880/**0.920** | 0.120/**0.080** | 0.000/0.000 | 8.084/**7.853** | 0.292/**0.321** |
| 5 | 0.866/**0.945** | 0.134/**0.055** | 0.000/0.000 | 8.040/**7.856** | 0.287/**0.342** |
| 6 | 0.868/**0.943** | 0.132/**0.057** | 0.000/0.000 | 8.014/**7.873** | 0.259/**0.345** |
| 7 | 0.858/**0.948** | 0.142/**0.052** | 0.000/0.000 | 8.026/**7.932** | 0.292/**0.350** |
| 8 | 0.828/**0.964** | 0.172/**0.036** | 0.000/0.000 | 8.043/**7.941** | 0.276/**0.362** |
| 9 | 0.868/**0.934** | 0.132/**0.066** | 0.000/0.000 | 8.019/**7.981** | 0.300/**0.343** |
| 10 | 0.896/**0.926** | 0.104/**0.074** | 0.000/0.000 | 8.016/**7.997** | 0.319/**0.343** |

**Table 4**
Evaluation of the robustness of CD−DRL−MP and the non-causal model under different disturbance attacks.

| $P^D$ | $\delta$ | Method | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|---|---|
| 0% | 0 | Non-causal | 0.890 | 0.110 | 0.000 | 8.089 | 0.310 |
| | | Causal | **0.938** | **0.062** | 0.000 | **7.948** | **0.342** |
| 10% | 0.5 | Non-causal | 0.870 | 0.130 | 0.000 | 8.144 | 0.295 |
| | | Causal | **0.925** | **0.075** | 0.000 | **7.973** | **0.334** |
| 10% | 1 | Non-causal | 0.800 | 0.200 | 0.000 | 8.258 | 0.255 |
| | | Causal | **0.910** | **0.090** | 0.000 | **8.030** | **0.323** |
| 20% | 0.5 | Non-causal | 0.870 | 0.130 | 0.000 | 8.124 | 0.297 |
| | | Causal | **0.935** | **0.065** | 0.000 | **7.979** | **0.334** |
| 20% | 1 | Non-causal | 0.780 | 0.220 | 0.000 | 8.317 | 0.237 |
| | | Causal | **0.925** | **0.075** | 0.000 | **8.089** | **0.329** |
| 30% | 0.5 | Non-causal | 0.835 | 0.165 | 0.000 | 8.162 | 0.275 |
| | | Causal | **0.885** | **0.115** | 0.000 | **8.003** | **0.306** |
| 30% | 1 | Non-causal | 0.705 | 0.295 | 0.000 | 8.363 | 0.197 |
| | | Causal | **0.865** | **0.135** | 0.000 | **8.097** | **0.292** |
| 40% | 0.5 | Non-causal | 0.860 | 0.140 | 0.000 | 8.222 | 0.288 |
| | | Causal | **0.940** | **0.060** | 0.000 | **8.033** | **0.330** |
| 40% | 1 | Non-causal | 0.670 | 0.330 | 0.000 | 8.407 | 0.173 |
| | | Causal | **0.900** | **0.100** | 0.000 | **8.164** | **0.309** |

Especially, CD−DRL−MP performs better than the non-causal model under all disturbance scenarios.

The reason is that by precisely blocking backdoor paths, a causal DRL model can learn a more robust control policy and generate better motion commands. Hence, we can conclude that by mitigating the detrimental effects of potential confounders, CD−DRL−MP can boost motion performance and improve robustness.

### 6.4. Generalizability of CD−DRL−MP

In this section, we evaluate the generalizability of CD−DRL−MP, i.e., whether a model trained in one environment is still available in other environments with different numbers of obstacles. Hence, we first train a CD−DRL−MP and a non-causal model in the environment with five obstacles. Subsequently, we evaluate the performance of either model under two situations: *seen environments* (i.e., five obstacles), where the model trained and tested in environments with the same number of obstacles, and *unseen environments* (i.e., 1–4 obstacles and 6–13 obstacles), where the model was trained in one environment and tested in other environments with different numbers of obstacles. Each test set consisted of 500 cases. The experimental results on the test sets are shown in Table 5. Rows 2 and 3 show the testing results of seen environments. Rows 4 and 5 show the average results for the two test sets of the unseen environments.

First, CD−DRL−MP shows better performance than the non-causal model in the seen environment (93.2% vs 90.2% for success rate). This indicates that CD−DRL−MP can be generalized well to the seen environment. Second, we can observe that CD−DRL−MP can still achieve better motion planning performance than the non-causal model in unseen testing environments. In contrast, the non-causal DRL model shows significant performance degradation when applied to unseen environments, e.g., 85.6% for the unseen environment and 90.2% for the seen environment in terms of success rate. From the results in Table 5, we can find that when applied to unseen environments, the non-causal model reduces the success rate by 5.099%, i.e., (0.902–0.856)/0.902, whereas CD−DRL−MP only reduces it by 0.751%, i.e., (0.932–0.935)/0.932. Table 5 empirically verifies that the policy learned by the proposed method can generalize to unseen testing environments. This is because CD−DRL−MP is based on causal relationships that have been proven to be invariant [9,40,41]. Hence, we can conclude that CD−DRL−MP can generalize well beyond the training environment and guarantee a good motion performance.

### 6.5. Compatibility analysis

In the sequel, we investigate the compatibility of the causal deconfounding module. We replace the motion planning module with other existing DRL-based motion planning methods. We select two well-established DRL methods: SARL [19] and Crit-LSTM [7]. We evaluate their performance under two simulation scenarios: *circle-crossing*, same as Section 6.2, and *square-crossing*, where all obstacles are randomly located on the left side or right side of a square and need to move to the opposite side. Four models were trained for each scenario: SARL, Crit-LSTM, CD-SARL, and CD-Crit-LSTM. Their hyper-parameters follow the configuration suggested by the original SARL [19] and Crit-LSTM [7]. All models are trained for 2000 episodes with a varying number of obstacles, varying from 1 to 10. Then, we test each model using 500 test cases with the number of obstacles varying from 1 to 10. Table 6 presents the results of the four models for each scenario.

First, as in CD−DRL−MP, the two causal models (i.e., CD-SARL and CD-Crit-LSTM) show better performance than their corresponding

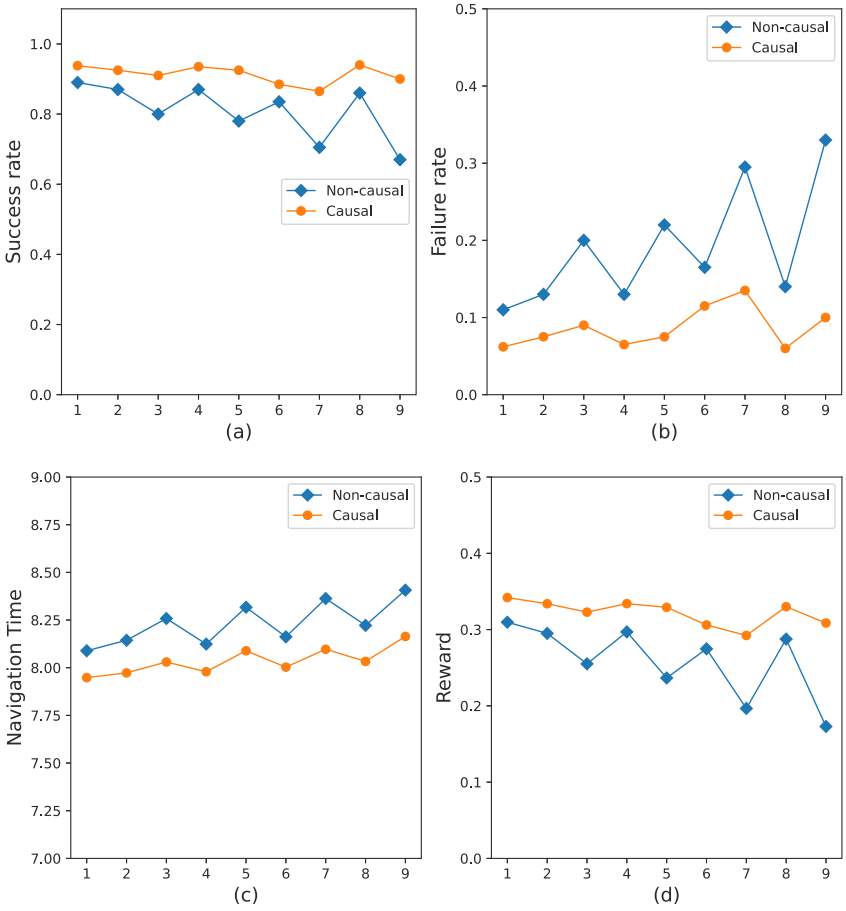



**Fig. 5.** Comparison of the performance for either model with the increase of attack strength.

**Table 5**
Evaluation of the generalizability of CD-DRL-MP and the non-causal model in the seen and unseen environments.

| Environment | Method | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|---|
| Seen | Non-causal | 0.902 | 0.098 | 0.000 | 8.311 | 0.302 |
| | Causal | **0.932** | **0.068** | 0.000 | **7.985** | **0.329** |
| Unseen | Non-causal | 0.856 | 0.144 | 0.000 | 8.319 | 0.283 |
| | Causal | **0.925** | **0.075** | 0.000 | **8.049** | **0.326** |

**Table 6**
Performance of the causal deconfounding module with different motion planning methods under two simulation scenarios.

| Scenario | Method | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|---|
| Circle-crossing | SARL | 0.876 | 0.062 | 0.062 | 8.443 | 0.307 |
| | CD-SARL | **0.928** | **0.072** | **0.000** | **7.939** | **0.333** |
| Square-crossing | SARL | 0.970 | 0.024 | 0.006 | 7.884 | 0.363 |
| | CD-SARL | **0.984** | **0.016** | **0.000** | **7.798** | **0.363** |
| Circle-crossing | Crit-LSTM | 0.940 | 0.060 | 0.000 | 7.934 | 0.341 |
| | CD-Crit-LSTM | **0.950** | **0.050** | 0.000 | **7.874** | **0.343** |
| Square-crossing | Crit-LSTM | 0.972 | 0.028 | 0.000 | 8.007 | 0.353 |
| | CD-Crit-LSTM | **0.980** | **0.020** | 0.000 | **7.768** | **0.361** |

non-causal models under all scenarios. This indicates that the causal deconfounding module can improve the performance of different baseline motion planning methods. Second, CD-SARL exhibits a more significant performance improvement, increasing from 87.6% to 92.8% for circle-crossing and from 97% to 98.4% for square-crossing. This indicates that SARL captures more spurious correlations during the training phase. Third, for each scenario, Crit-LSTM achieves a higher performance than SARL. This is because Crit-LSTM deals with obstacles' states according to their criticality, which can capture more useful high-level semantic information. Consequently, we can conclude that the proposed causal deconfounding module is universal and compatible with other existing DRL-based motion-planning methods.

### 6.6. Ablation study

Finally, to justify the design of the causal method, we further conduct ablation studies. We take the causal model as the baseline and

**Table 7**
Ablation studies for the design of the proposed causal deconfounding DRL method (non-causal/causal).

| Metric | Causal Method | Variant1 | Variant2 | Variant3 | Variant4 | Variant5 | Variant6 |
|---|---|---|---|---|---|---|---|
| Success | **0.946** | 0.935 | 0.899 | 0.929 | 0.902 | 0.931 | 0.907 |
| Collision | **0.054** | 0.065 | 0.097 | 0.071 | 0.077 | 0.069 | 0.061 |
| Timeout | **0.000** | 0.000 | 0.003 | 0.000 | 0.021 | 0.000 | 0.033 |
| NavTime | **7.919** | 8.089 | 8.058 | 7.988 | 8.155 | 7.928 | 8.035 |
| Reward | **0.344** | 0.330 | 0.314 | 0.338 | 0.319 | 0.335 | 0.333 |

study six kinds of ablation: (1) *Variant1* ($\mathbf{S}^{\pi}_{t,causal} \backslash \mathbf{S}^{\pi}_{t-1}$), where a model is trained based on the state set $\{\mathbf{S}^{\pi}_{t}, \mathbf{v}_{t-1}\}$ without considering the previous states of the robot and obstacles; (2) *Variant2* ($\mathbf{S}^{\pi}_{t,causal} \backslash \mathbf{v}_{t-1}$), where the state set is replaced by $\{\mathbf{S}^{\pi}_{t}, \mathbf{S}^{\pi}_{t-1}\}$ without considering the robot's previous velocity; (3) *Variant3* ($\mathbf{S}^{\pi}_{t,causal} \backslash \mathbf{S}^{\pi}_{t}$), where the model only considers the previous state and action $\{\mathbf{S}^{\pi}_{t-1}, \mathbf{v}_{t-1}\}$; (4) *Variant4* ($\mathbf{S}^{\pi}_{t,causal} \backslash \{\mathbf{S}^{\pi}_{t-1}, \mathbf{v}_{t-1}\}$), where no previous information is considered, i.e., the non-causal model. (5) *Variant5* ($\mathbf{S}^{\pi}_{t,causal} \cup \mathbf{v}_{t-2}$), where the state set is replaced by $\{\mathbf{S}^{\pi}_{t}, \mathbf{S}^{\pi}_{t-1}, \mathbf{v}_{t-1}, \mathbf{v}_{t-2}\}$; (6) *Variant6* ($\mathbf{S}^{\pi}_{t,causal} \cup \mathbf{S}^{\pi}_{t-2}$), where the model is trained based on state set $\{\mathbf{S}^{\pi}_{t}, \mathbf{S}^{\pi}_{t-1}, \mathbf{v}_{t-1}, \mathbf{S}^{\pi}_{t-2}\}$. We use `CD-DRL-MP`, CD-SARL, and CD-Crit-LSTM as the baseline causal models and compared them with their corresponding variants. Other experimental settings are the same as those provided in Section 6.5.

Table 7 presents the average results for the different variants over the three causal models. The following conclusions can be drawn from the table. (1) By comparing `CD-DRL-MP` with its variants, we can observe that precisely blocking all backdoor paths can achieve better performance. (2) Removing any element in $\mathbf{S}^{\pi}_{t,causal}$ (i.e., *Variant1*, *Variant2*, and *Variant3*) causes a significant performance degradation. This implies that the proposed causal state representation is the minimal sufficient set to mitigate the effect of potential confounders. (3) Comparison of *Variant1*, *Variant2*, and the non-causal method (i.e., *Variant4*) illustrates that blocking partial backdoor paths can still improve motion planning performance. This means that embedding causal relationships within DRL is an effective way to promote DRL-based motion planning methods. (4) Comparison of *Variant1*, *Variant5*, and *Variant6* illustrates that adding more information into the deconfounding set cannot boost the performance. The ablation results also justify the proposed $\mathbf{S}^{\pi}_{t,causal}$ is the minimal sufficient deconfounding set.

In conclusion, by enhancing DRL motion planning methods with deconfounding capabilities through a causal framework, our proposed approach significantly boosts the performance of DRL methods in motion planning while also improving their robustness and generalizability.

### 6.7. Experiments in Gazebo

In this section, we further evaluate the proposed method using *RotorS* [42], which is a high-fidelity Gazebo simulator. Gazebo is a mainstream open-source platform that accurately models and reflects the physical characteristics of real-world robots. In our experiments, the *AscTec Firefly* drones are developed to simulate the robot and obstacles in an environment. All drones can communicate using the topic subscription and publication mechanism in ROS. Each drone is simulated with a ROS node deployed on an Ubuntu 18.04.6 LTS system with ROS *Melodic*, to execute the motion planning algorithms and generate motion commands. The safety radius of each drone is 0.3 m, and the robot drone is required to move from $[0, -4]$ to $[0, 4]$. All the simulation videos are available at https://sites.google.com/view/cd-drl-mp.

Fig. 6 shows a scenario of the robot and two obstacles. Fig. 6(a) presents the initial snapshot of the scenario. As shown in Fig. 6(a), the two obstacles must move from the robot's left to the right side. The robot must avoid collisions with these obstacles while moving towards its target position. Fig. 8(a) shows the actual trajectories of the three drones. Finally, we can see that the robot safely reaches its target position, as shown in Fig. 6(b).

Fig. 7(a) shows another scenario involving a robot and three obstacles. Fig. 8(b) shows the trajectories traveled by all drones. The target state of this scenario is shown in Fig. 7(b). We can see that the proposed method can help the robot avoid potential collisions with obstacles and navigate the robot to its target position.

**Table 8**
Comparison of the performance between the proposed causal deconfounding (CD) module and other causal methods.

| Method | Success | Collision | Timeout | NavTime | Reward |
|---|---|---|---|---|---|
| CD | **0.938** | **0.062** | **0.000** | **8.084** | **0.344** |
| OREA | 0.888 | 0.101 | 0.011 | 8.409 | 0.300 |
| iCaRL | 0.882 | 0.118 | 0.000 | 8.312 | 0.290 |

### 6.8. Comparison with other causal methods

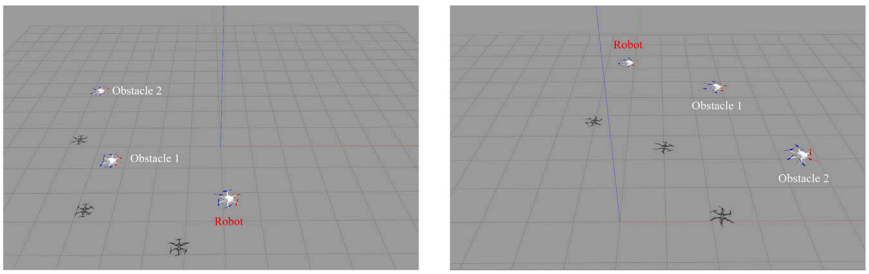
Finally, the proposed causal deconfounding module was compared with other existing causal methods. Specifically, the object-aware regularization (OREA) method [43] and the invariant causal representation learning (iCaRL) method [9] were selected as baselines. The OREA method uses a regularization technique to mitigate the causal confounder problem, whereas the iCaRL method constructs an invariant predictor to improve the generalizability of the trained policy. For each causal method, we implement them using LSTM-RL [6] as the motion planning module.

Table 8 lists the results of three implemented methods over the ten test sets. Each test set contained 500 test cases with varying number of obstacles ranging from 1 to 10. From the results, we can find that our proposed causal deconfounding method outperforms the existing baselines, achieving a higher success rate (93.8%). This implies that the proposed method is more effective in mitigating confounders in motion planning than the other two methods. This is because our method relies on strict causal modeling for the motion planning task and utilizes a well-designed deconfounding strategy. The results also indicate the good capability of causal inference in learning motion planning policies, which has been largely ignored by most previous DRL methods.

## 7. Conclusion

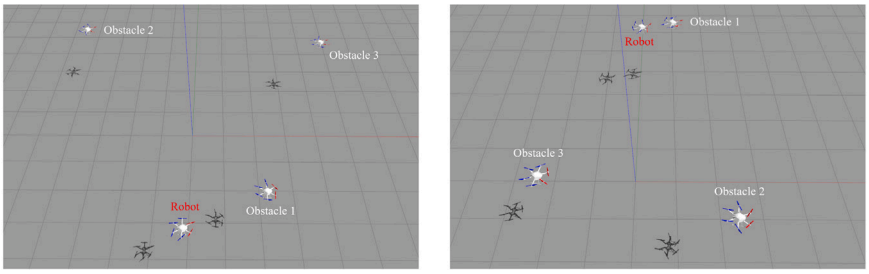
In this paper, we propose the first causal DRL method for motion planning in mobile robots. By blocking all backdoor paths, we propose learning the causal relationships between states and actions, while mitigating the detrimental effects of potential confounders. First, we use a structural causal model to formalize the temporal causal relationships of the motion planning task. Second, based on the built causal model, we explain why a non-causal DRL method failed to learn the expected causal effects. Third, we propose a minimal sufficient deconfounding set and prove it can block all backdoor paths from states to actions in the causal model. Finally, by leveraging the deconfounding set, we propose a causal DRL method for motion planning, `CD-DRL-MP`, for learning robust and generalizable control policies. Comprehensive simulation experiments demonstrate that `CD-DRL-MP` can deal with confounders in historical information, improve motion planning performance significantly, as well as guarantee good robustness and generalizability. Our findings suggest that incorporating causal information can enhance the motion planning task.

This study is the first attempt to empower DRL motion planning methods with causal deconfounding capabilities. One limitation of



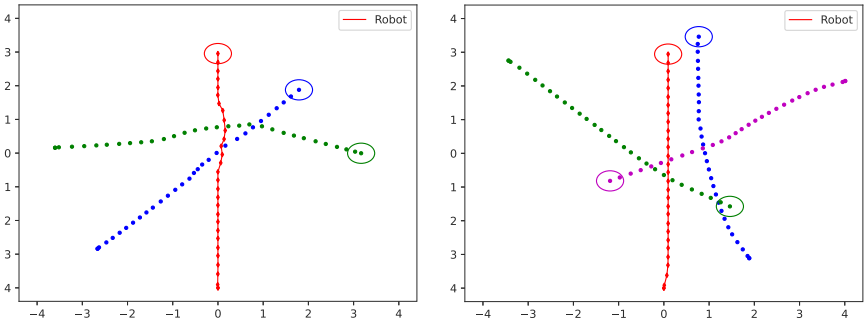

(a) Initial state. (b) Target state.

**Fig. 6.** *RotorS* experiments with two obstacles.




(a) Initial state. (b) Target state.

**Fig. 7.** *RotorS* experiments with three obstacles.




(a) Traveled trajectory of Figure 6. (b) Traveled trajectory of Figure 7.

**Fig. 8.** Trajectories traveled by the robot and obstacles in the *RotorS* Simulator.

our method is its focus on cause–effect relationships among high-level variables, while the semantic information within these variables also warrants exploration. For example, identifying the key semantic information that results in backdoor paths from a state variable can contribute to more fine-grained mitigation of spurious correlations. We believe that encouraging the DRL policy to address key causal semantic information is a promising direction for addressing spurious correlations. Another limitation is that real-world environments may have other potential confounders, such as weather and map structure. In this paper, we mainly focus on mitigating confounders in the historical information by assuming all other aspects are fixed. Hence, we verify the effectiveness and efficiency of our method via theoretical analysis and simulation-based evaluation. In the future, we plan to implement and test the proposed method on real-world robotic platforms to explore its ability to mitigate various confounders.

## CRediT authorship contribution statement

**Wenbing Tang:** Writing – original draft, Visualization, Resources, Methodology. **Fenghua Wu:** Validation, Software, Investigation. **Shang-wei Lin:** Writing – review & editing. **Zuohua Ding:** Writing – review & editing, Resources, Methodology. **Jing Liu:** Writing – review & editing, Supervision. **Yang Liu:** Methodology. **Jifeng He:** Supervision, Methodology.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgments

We would like to thank the reviewers and editors for their constructive comments. Jing Liu would like to thank the support of the National Key Research and Development Program of China under Grant 2022YFC3302600.

This work was supported in part by the National Research Foundation, Singapore, and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-GC-2023-008), NRF Investigatorship NRF-NRFI06-2020-0001, Academic Research Fund Tier 2 by Ministry of Education in Singapore under Grant No. MOE-T2EP20120-0004, Natural Science Foundation of China under Grant No. 62132014, Zhejiang Provincial Key Research and Development Program of China under Grant 2022C01045, and Science Foundation of Zhejiang Sci-Tech University (ZSTU) Under Grant No. XJ2024000701.

## References

[1] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, et al., Motion planning for autonomous driving: The state of the art and future perspectives, IEEE Trans. Intell. Veh. (2023).

[2] Z. Li, H. Liang, H. Wang, X. Zheng, J. Wang, P. Zhou, A multi-modal vehicle trajectory prediction framework via conditional diffusion model: A coarse-to-fine approach, Knowl.-Based Syst. 280 (2023) 110990.

[3] L. Antonyshyn, J. Silveira, S. Givigi, J. Marshall, Multiple mobile robot task and motion planning: A survey, ACM Comput. Surv. 55 (10) (2023) 1–35.

[4] C. Zhou, B. Huang, P. Fränti, Representation learning and reinforcement learning for dynamic complex motion planning system, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[5] S. Matsuzaki, Y. Hasegawa, Learning crowd-aware robot navigation from challenging environments via distributed deep reinforcement learning, in: 2022 International Conference on Robotics and Automation, ICRA, IEEE, 2022, pp. 4730–4736.

[6] M. Everett, Y.F. Chen, J.P. How, Motion planning among dynamic, decision-making agents with deep reinforcement learning, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 3052–3059.

[7] L. Xu, F. Wu, Y. Zhou, H. Hu, Z. Ding, Y. Liu, Criticality-guided deep reinforcement learning for motion planning, in: 2021 China Automation Congress, CAC, IEEE, 2021, pp. 3378–3383.

[8] K. Ruan, X. Di, Learning human driving behaviors with sequential causal imitation learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, No. 4, 2022, pp. 4583–4592.

[9] C. Lu, J.M. Hernández-Lobato, B. Schölkopf, Invariant causal representation learning for generalization in imitation and reinforcement learning, in: ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality, 2022.

[10] K. Kuang, R. Xiong, P. Cui, S. Athey, B. Li, Stable prediction with model misspecification and agnostic distribution shift, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 04, 2020, pp. 4485–4492.

[11] B. Huang, C. Lu, L. Leqi, J.M. Hernández-Lobato, C. Glymour, B. Schölkopf, K. Zhang, Action-sufficient state representation learning for control with structural constraints, in: International Conference on Machine Learning, PMLR, 2022, pp. 9260–9279.

[12] C. Li, G. Feng, Y. Li, R. Liu, Q. Miao, L. Chang, DiffTAD: Denoising diffusion probabilistic models for vehicle trajectory anomaly detection, Knowl.-Based Syst. 286 (2024) 111387.

[13] W. Tang, Y. Zhou, H. Sun, Y. Zhang, Y. Liu, Z. Ding, J. Liu, J. He, Gan-based robust motion planning for mobile robots against localization attacks, IEEE Robot. Autom. Lett. 8 (3) (2023) 1603–1610.

[14] X. Zhang, J. Wang, Y. Fang, J. Yuan, Multilevel humanlike motion planning for mobile robots in complex indoor environments, IEEE Trans. Autom. Sci. Eng. 16 (3) (2018) 1244–1258.

[15] Y. Ji, L. Ni, C. Zhao, C. Lei, Y. Du, W. Wang, TriPField: A 3D potential field model and its applications to local path planning of autonomous vehicles, IEEE Trans. Intell. Transp. Syst. 24 (3) (2023) 3541–3554.

[16] L. Chen, X. Hu, B. Tang, Y. Cheng, Conditional DQN-based motion planning with fuzzy logic for autonomous driving, IEEE Trans. Intell. Transp. Syst. 23 (4) (2020) 2966–2977.

[17] M. Cai, S. Xiao, Z. Li, Z. Kan, Optimal probabilistic motion planning with potential infeasible LTL constraints, IEEE Trans. Autom. Control 68 (1) (2021) 301–316.

[18] J. Van Den Berg, S.J. Guy, M. Lin, D. Manocha, Reciprocal n-body collision avoidance, in: Robotics Research: The 14th International Symposium ISRR, Springer, 2011, pp. 3–19.

[19] C. Chen, Y. Liu, S. Kreiss, A. Alahi, Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 6015–6022.

[20] P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, J. Pan, Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 6252–6259.

[21] K. Wu, H. Wang, M.A. Esfahani, S. Yuan, Learn to navigate autonomously through deep reinforcement learning, IEEE Trans. Ind. Electron. 69 (5) (2021) 5342–5352.

[22] S.H. Semnani, H. Liu, M. Everett, A. De Ruiter, J.P. How, Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning, IEEE Robot. Autom. Lett. 5 (2) (2020) 3221–3226.

[23] Q. Zhou, Y. Lian, J. Wu, M. Zhu, H. Wang, J. Cao, An optimized Q-learning algorithm for mobile robot local path planning, Knowl.-Based Syst. 286 (2024) 111400.

[24] Y. Yao, J. Zhao, Z. Li, X. Cheng, L. Wu, Jamming and eavesdropping defense scheme based on deep reinforcement learning in autonomous vehicle networks, IEEE Trans. Inf. Forensics Secur. 18 (2023) 1211–1224.

[25] T.D. Duong, Q. Li, G. Xu, Causality-based counterfactual explanation for classification models, Knowl.-Based Syst. (2024) 112200.

[26] M. Chen, H. Wang, R. Wang, Y. Peng, H. Zhang, CDRM: Causal disentangled representation learning for missing data, Knowl.-Based Syst. (2024) 112079.

[27] P. De Haan, D. Jayaraman, S. Levine, Causal confusion in imitation learning, Adv. Neural Inf. Process. Syst. 32 (2019).

[28] J. Tien, J.Z.-Y. He, Z. Erickson, A.D. Dragan, D.S. Brown, Causal confusion and reward misidentification in preference-based reward learning, 2022, arXiv preprint arXiv:2204.06601.

[29] J. Zhang, D. Kumor, E. Bareinboim, Causal imitation learning with unobserved confounders, Adv. Neural Inf. Process. Syst. 33 (2020) 12263–12274.

[30] D. Kumor, J. Zhang, E. Bareinboim, Sequential causal imitation learning with unobserved confounders, Adv. Neural Inf. Process. Syst. 34 (2021) 14669–14680.

[31] K. Ruan, J. Zhang, X. Di, E. Bareinboim, Causal imitation learning via inverse reinforcement learning, in: The Eleventh International Conference on Learning Representations, 2022.

[32] J. Li, K. Kuang, B. Wang, F. Liu, L. Chen, C. Fan, F. Wu, J. Xiao, Deconfounded value decomposition for multi-agent reinforcement learning, in: International Conference on Machine Learning, PMLR, 2022, pp. 12843–12856.

[33] I. Bica, D. Jarrett, M. van der Schaar, Invariant causal imitation learning for generalizable policies, Adv. Neural Inf. Process. Syst. 34 (2021) 3952–3964.

[34] J. Pearl, Causality, Cambridge University Press, 2009.

[35] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[36] C. Mao, K. Xia, J. Wang, H. Wang, J. Yang, E. Bareinboim, C. Vondrick, Causal transportability for visual recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 7521–7531.

[37] X. He, Y. Zhang, F. Feng, C. Song, L. Yi, G. Ling, Y. Zhang, Addressing confounding feature issue for causal recommendation, ACM Trans. Inf. Syst. 41 (3) (2023) 1–23.

[38] F. Lv, J. Liang, S. Li, B. Zang, C.H. Liu, Z. Wang, D. Liu, Causality inspired representation learning for domain generalization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8046–8056.

[39] N.A. Heckert, J.J. Filliben, Nist/Sematech e-Handbook of Statistical Methods; Chapter 1: Exploratory Data Analysis, N. Alan Heckert, James J. Filliben, 2003.

[40] Y.-F. Zhang, Z. Zhang, D. Li, Z. Jia, L. Wang, T. Tan, Learning domain invariant representations for generalizable person re-identification, IEEE Trans. Image Process. 32 (2022) 509–523.

[41] C. Xu, C. Liu, X. Sun, S. Yang, Y. Wang, C. Wang, Y. Fu, PatchMix augmentation to identify causal features in few-shot learning, IEEE Trans. Pattern Anal. Mach. Intell. (2022).

[42] F. Furrer, M. Burri, M. Achtelik, R. Siegwart, Rotors—a modular gazebo mav simulator framework, in: Robot Operating System (ROS) the Complete Reference (Volume 1), Springer, 2016, pp. 595–625.

[43] J. Park, Y. Seo, C. Liu, L. Zhao, T. Qin, J. Shin, T.-Y. Liu, Object-aware regularization for addressing causal confusion in imitation learning, Adv. Neural Inf. Process. Syst. 34 (2021) 3029–3042.