# SVCE: Shapley Value Guided Counterfactual Explanation for Machine Learning-Based Autonomous Driving

Meng Li, Hengyang Sun, Yanjun Huang, and Hong Chen, *Fellow, IEEE*

***Abstract*— The explainability of complex machine-learning models is becoming increasingly significant in safety-critical domains such as autonomous driving. In this context, counterfactual explanation (CE), as an effective explainability method in explainable artificial intelligence, plays an important role. It aims to identify minimal alterations to input that can change the model's output, thereby revealing key factors influencing model decisions. However, generating counterfactual samples might involve manually selecting input features, potentially leading to suboptimal and biased explanations. This study introduces a feature contribution guided CE generation framework to address this issue. Our method utilizes feature contributions based on Shapley values to guide the model's focus on the most influential features. This enables end-users to quickly pinpoint the search direction in generating CEs (e.g., prioritizing the most critical features) and producing representative CEs. To comprehensively evaluate our method, we conducted experimental validation on two representative machine learning models: autonomous driving decision-making using Deep Q-Network and lane-changing prediction using deep learning. In addition, we conducted a user-centered study to evaluate the practical applicability of the SVCE in autonomous driving scenarios, which serves as a crucial validation of the presented SVCE. The results show that SVCE can help users understand and diagnose the model.**

***Index Terms*— Explainable artificial intelligence, shapley value, machine learning, counterfactual explanation, autonomous driving.**

## I. Introduction

MACHINE learning models are being increasingly utilized, especially within safety-critical domains such as autonomous driving [1], [2], [3], [4]. However, these models are often perceived as black boxes, sparking significant concerns [5]. Researchers have embarked on developing explainability methods to enhance the credibility of these models. Among the most emerging is the counterfactual explanation (CE) method, which aims to delve into the model's decision-making process [6], [7]. Given a decision model and an input query, the CE method attempts to find a data point with a minimal difference from the query but sufficient to alter the model's decision. By contrasting the differences between the query and the explanation, users can infer in reverse to understand the key factors and boundaries the model relies on. In the context of autonomous driving, counterfactual explanations can enhance our understanding of why autonomous driving models choose specific actions and diagnose the model [8], [9].

In recent years, numerous methods have been proposed to generate CEs. Perturbation-based techniques modify the input data by adding or removing specific features [10]. In contrast, gradient-based methods utilize gradients to identify and alter features in the input data [6]. However, these methods involve manually selecting input features, which may result in suboptimal and biased explanations [11]. The main objective of this research is to generate CEs that do not rely on manual feature specification, with a particular focus on scenarios in autonomous driving.

In response to these prevailing challenges, we introduce Shapley Value Guided Counterfactual Explanation (SVCE), which naturally determines the features to be altered for generating CEs using Shapley values. We employ it in two representative machine-learning models to comprehensively evaluate our approach. We first apply SVCE to a Deep Q-Network (DQN)-based autonomous driving decision model and subsequently integrate it into a lane-change prediction model based on deep learning. For instance, Fig. 1 displays the CEs discovered by SVCE. Given a query sample ($top\ left$) and a decision system, SVCE alters the most influential features ($bottom\ left$) based on the feature contributions represented by the Shapley values ($top\ right$) and thus results in a changed decision. By presenting CEs at the bottom, users can perform reverse inference to understand the key factors influencing the model's decision and diagnose the model.

The main contributions are threefold:

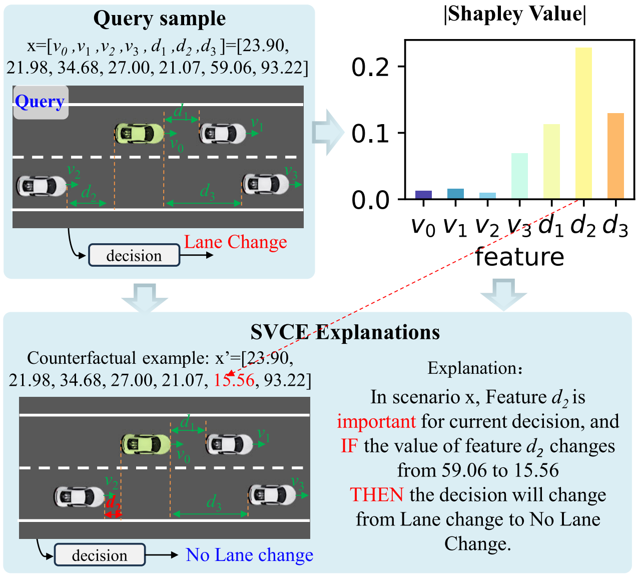
- We have tackled the issue of manually generating CEs in decision models through Shapley value, specifically targeting scenarios in autonomous driving.
- We perform a comprehensive analysis of SVCE employing the autonomous driving models. The results provide empirical evidence for insightful model explanations and discovering model biases.
- We conduct a user-centric study to evaluate the practical utility of SVCE, providing evidence that SVCE can

Manuscript received 7 November 2023; revised 7 March 2024; accepted 22 April 2024. This work was supported by the National Natural Science Foundation of China, Joint Fund for Innovative Enterprise Development, under Grant U23B2061. The Associate Editor for this article was J. Catalão. *(Corresponding author: Yanjun Huang.)*

Meng Li and Hong Chen are with the Department of Control Science and Engineering, Tongji University, Shanghai 200092, China (e-mail: 2010460@tongji.edu.cn; chenhong2019@tongji.edu.cn).

Hengyang Sun and Yanjun Huang are with the Clean Energy Automotive Engineering Center, Tongji University, Shanghai 200092, China (e-mail: 1852014@tongji.edu.cn; yanjun_huang@tongji.edu.cn).

Digital Object Identifier 10.1109/TITS.2024.3393634


Query sample
x=[$v_0$,$v_1$,$v_2$,$v_3$,$d_1$,$d_2$,$d_3$]=[23.90, 21.98, 34.68, 27.00, 21.07, 59.06, 93.22]
Query
$d_1$
$v_1$
$v_0$
$v_2$
$v_3$
$d_2$
$d_3$
decision
Lane Change
|Shapley Value|
0.2
0.1
0.0
$v_0$ $v_1$ $v_2$ $v_3$ $d_1$ $d_2$ $d_3$
feature
SVCE Explanations
Counterfactual example: x'=[23.90, 21.98, 34.68, 27.00, 21.07, 15.56, 93.22]
decision
No Lane change

Explanation:
In scenario x, Feature $d_2$ is important for current decision, and IF the value of feature $d_2$ changes from 59.06 to 15.56 THEN the decision will change from Lane change to No Lane Change.

Fig. 1. **Counterfactual explanation generated by SVCE**. Given an original query sample (*top left*), SACE modifies the top $m$ (here $m$ sets to 1) influential features based on shapley values (*top right*) to generate counterfactual samples sufficient to alter the decision (*bottom, left*). The predictions made by the model are reported below the original and counterfactual sample. The explanations represented at the bottom (*right*).

enhance users' understanding and diagnostic capabilities of the model.

The remainder of this paper is organized as follows. Section II briefly reviews the related work on explainable AI, Shapley values, and CEs. Section III describes how SVCE steers the generation of CEs. Section IV presents the evaluation results and discusses their implications. Finally, we conclude this work in Section V.

## II. Background and Related Works

### A. Local Explanations

Most machine-learning models are not designed with explainability in mind, with only a few exceptions [12], [13]. In safety-critical applications, such as autonomous driving or medical imaging, explanations are essential for legal liability and enhancing end-user trust [14], [15], [16], [17]. This has led to a post-hoc interest in explaining trained models, which can help analyze special cases, understand model failures, and discover biases [12], [18]. Post-hoc explanations can be global, providing an overall view of the primary decision factors of a model. In this paper, we focus on local explanation methods, which aim to understand the behavior of a model on specific inputs (possibly structured features and pixel features) [19]. The most representative local explanation methods include Shapley value-based feature attribution explanations and CEs.

### B. Shapley Values

Shapley values are originally used for value distribution among players in a cooperative game. Today, Shapley values have become a popular feature attribution explanation method, widely used for model explanation and feature importance evaluation, aiming to deepen our understanding of the contribution of each feature to the model prediction. On the one hand, Shapley values can measure the feature contribution of models developed on tabular data [3], [20], [21], which helps us understand the impact of each feature on the overall model performance. On the other hand, they generate saliency maps that emphasize the pixels or areas that influence the model decision [22], [23], [24]. These saliency maps help visually understand the model's decision basis for different inputs, providing visual explanation support. However, although Shapley values have apparent advantages in model explanation, they also have some limitations. In particular, feature attribution methods represented by Shapley values usually only provide information about the features or areas of interest to the model without clearly indicating their impact on the model decision boundary. Therefore, in some cases, more fine-grained local explanation methods may need to supplement Shapley value analysis further to provide a more comprehensive model explanation. In this direction, CE methods have become a significant advancement.

### C. Counterfactual Explanations

CEs are defined for a given decision model and query input, with the core idea being to find the most minor but meaningful modification to the query that is sufficient to change the model's decision [6]. Perturbation-based techniques involve altering input data by introducing changes such as adding or removing specific features [10]. Conversely, gradient-based methods rely on the model's gradient to identify and subsequently modify features within the input data [6]. Nonetheless, it is essential to note that both methodologies necessitate the manual selection of input features. This manual intervention can introduce potential issues, including the risk of suboptimal and biased explanations [11]. Recent research has integrated Shapley values and CEs [25], producing counterfactual samples based on feature contributions. However, these methods, which often involve removing important features to generate counterfactual samples, fail to provide insights into the actual decision boundaries of the model, specifically, how changes in features affect the model's decision. To address these issues, we propose SVCE, which naturally determines the features to be altered for generating CEs using Shapley values. In addition, we demonstrate the potential of the SVCE method in explaining and diagnosing safety-critical autonomous driving models.

## III. Shapley Value Guided Counterfactual Explanation

This section proposes SVCE, a method for automated generating counterfactual examples based on Shapley values. We first formally describe our method and objectives in Section III-A. Subsequently, Section III-B discusses the Shapley value-based feature attribution. Lastly, in Section III-C, we elaborate on how to generate the feature attribution-targeted CEs. The implementation of SVCE is depicted in Fig 2.

### A. Problem Description and Optimization Objectives

For a decision model $f$, the objective of generating CEs is to find a counterfactual instance $x'$ that is as close as possible to the query instance $x$ to be explained such that $c(f(x)) \neq c(f(x'))$, where $c(\cdot)$ represents an equation that maps the output of $f(x)$ to the selected category. This can be formally defined as follows:

$$\arg\min_{x' \in X} L_{\text{dist}}(x, x') \quad s.t.\ c(f(x)) \neq c(f(x')). \tag{1}$$

However, this task faces a key issue: how to define distance metric $L_{dist}(x, x')$. To address this issue and generate non-trivial CEs in a more refined manner, we assign a contribution (or importance) degree to each feature for the query sample $x$. Specifically, our method is guided by a feature contribution model $\phi(f, x)$. It provides the contribution of each feature in the instance $x$, guiding the generation of a counterfactual instance $x'$. The process mentioned above can be transformed into the following optimization problem:

$$\arg\min_{x' \in X} L_{\text{dist}}\left(x, x'\right) \odot \phi(f, x) \ \ s.t.\ c(f(x)) \neq c(f(x')), \tag{2}$$

where $\odot$ denotes the impact function of $\phi(f, x)$ on $L_{\text{dist}}(x, x')$. This will be detailed in Sections III-B and III-C.

*Remark 1:* Eq (2) provides a formalized expression for counterfactual explanations in decision or classification models. It can be readily adapted to linear regression models by modifying the constraint condition from $c(f(x)) \neq c(f(x'))$ to $f(x) \neq f(x')$.

### B. Shapley Value-Based Feature Attributions

Feature contribution model $\phi(f, x)$ can be obtained through feature attribution method. However, most feature attribution methods often fail to ensure efficiency, i.e., the sum of local feature attributions does not equal the model's prediction. Therefore, we choose a flexible, accurate attribution model, Shapley values, used in cooperative game theory to calculate player contributions. Shapley values uniquely satisfy efficiency, symmetry, dummy player, and additivity/linearity properties [26].

To calculate the contribution of the $i$-th feature $x_i$ to the model $f$, we need to obtain all subsets $S \subseteq N$ which include all features except $x_i$, where $N$ is the set of all features. Then, we compute the model $f(S \cup i)$ that includes the $i$-th feature and the model $f(S)$ that does not include the $i$-th feature. The difference between the two models is expressed as $f(S \cup i) - f(S)$, and the Shapley value of $x_i$ is computed as the weighted average of all possible differences, that is:

$$\phi_i = \frac{1}{|N|!} \sum_{S \subseteq N \setminus i} \kappa(N, S)\left(f(S \cup i) - f(S)\right), \tag{3}$$

where $\kappa(N, S) = |S|!(|N| - |S| - 1)!$. To compute $f(S)$, we must define each feature's presence or absence. The presence of feature $i$ will mean that the model is evaluated with the observed value $x_i$. For absent features, a straightforward way is to replace their values with baseline samples $x_i^b$. That is, if feature $i$ is not present, the value of this feature is set to $x_i^b$. Therefore, we can express $f(S)$ as $f\left(\tau\left(x, x^b, S\right)\right)$, such that if $i \in S$, then $\tau\left(x, x^b, S\right)_i$ equals $x_i$, otherwise it is $x_i^b$. To mitigate the instability in Shapley value calculations caused by individual base values, it's common practice to randomly sample from multiple base values and compute the expected value [27].

### C. Feature Attribution-Targeted CEs

We introduce a setting that allows for more fine-grained control over the generation of CEs. In this setting, the user can specify a set of features directly driven by the Shapley value. For example, in Fig. 2, if the two most crucial features are $v_0$ and $v_1$, the SVCE can modify only these features and produce corresponding CEs. This selection can be done without relying on manual selection to study the impact of different combinations of essential features on the target model behavior.

In practice, given a feature vector $\phi$ (as in (3)):

$$\phi = \left[\phi_1 \quad \cdots \quad \phi_i \quad \cdots \quad \phi_n\right], \tag{4}$$

where each feature contribution may be positive, indicating that the feature increases the value of the prediction, or it may be negative, indicating that the feature decreases the value of the prediction. However, our goal is to measure the importance of each feature. Therefore, we take the absolute value of each element $\phi_i$:

$$\phi' = [|\phi_1|, \cdots, |\phi_i|, \cdots, |\phi_n|]. \tag{5}$$

Then, given an integer $m \leq n$, which is used to control the number of important features selected for modification query sample, this is achieved by defining a binary mask vector $z$:

$$\begin{aligned} z &= \left[z_1 \quad \cdots \quad z_i \quad \cdots \quad z_n\right] \\ z_i &= \begin{cases} 1, & \text{if } i \in I = \arg\max\limits_{I} \left(\{\phi_i'\}_{i=1}^n, m\right) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad , \tag{6}$$

where $I$ denotes the set of indices corresponding to the top $m$ elements in $\phi'$. By combining (2), (5) and (6), we construct an optimization objective for weighted counterfactual samples:

$$\arg\min_{x' \in X} \left( z \cdot \phi' \cdot \left(x - x'\right)^2 = \sqrt{\sum_{i=1}^{n} \left(z_i \cdot \phi_i' \cdot \left(x_i - x_i'\right)\right)^2} \right). \\ s.t.\ c(f(x)) \neq c(f(x')) \text{ and } x_i' = x_i \text{ if } z_i = 0 \tag{7}$$

In the appendix, we present the pseudocode for the neural network-specific SVCE as Algorithm 1 and the model-agnostic SVCE as Algorithm 2.

## IV. Experiments

Data-driven autonomous driving models typically exhibit opacity, making it challenging to understand the model's decision-making process. Therefore, we will utilize the SVCE to provide model explanations and assist in diagnosing potential flaws in the model.

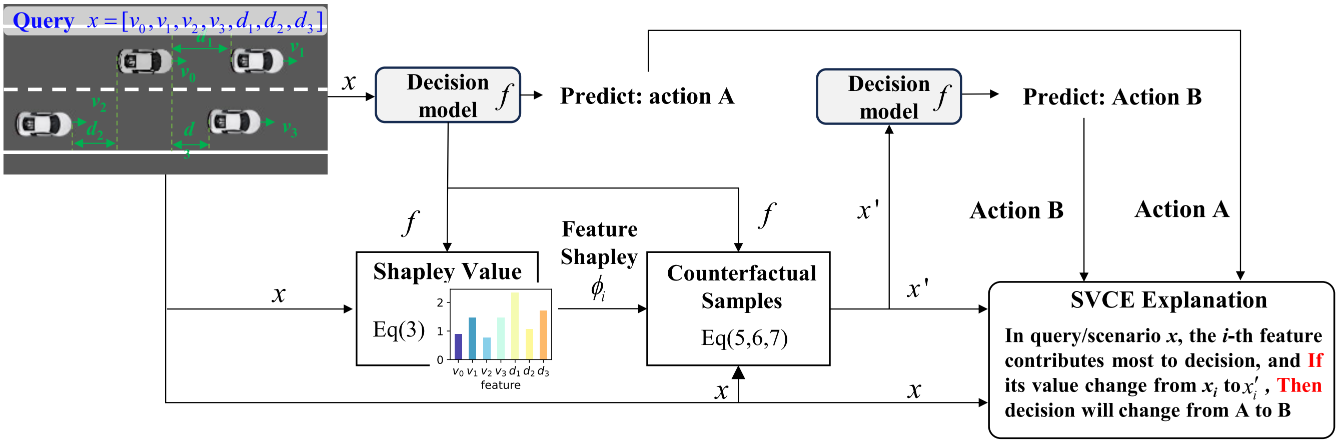


Fig. 2. **An Overview of SVCE**. The process begins with a query sample (*top left*), which passes through the shapley value module to estimate the feature contributions of the sample. Subsequently, a counterfactual sample $x'$ is constructed based on these contributions, leading to a decision modification. Finally, the explanation module presents the generation of a CE. Users can perform reverse inference to understand the key factors influencing the model's decision.

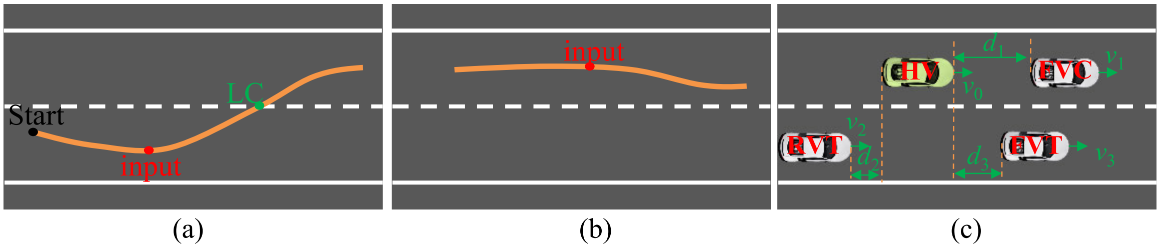


Fig. 3. Data extraction and feature selection for Lane-Change/Non-Lane-Change [3]. (a) Trajectory of vehicle No. 58 is recorded as a lane change. (b) Trajectory of vehicle No. 17 is recorded as no lane change. (c) Feature space.

### A. Experimental Design and Training Details

We evaluated our method on two representative machine-learning models. Firstly, we conducted experiments on an autonomous following vehicle model based on DQN. Then, we applied our method to a deep learning-based lane-change prediction model.

*1) Autonomous Following Vehicle Model Based on DQN:* We set up the autonomous driving environment in the Carla simulator [28]. The longitudinal decision-making scenario considers the difference between the leading vehicle concerning the speed of the ego vehicle, denoted as $\Delta v$, the speed of the ego vehicle, $v$, and the distance between the leading vehicle and the ego vehicle, $d$. We represent the state as $x = [\Delta v, v, d]$. The decision in the longitudinal direction is represented as $a = [a_{dec}, a_{idle}, a_{acc}]$, where $a_{idle}$ represents maintaining the previous decision, and $a_{dec}$ ($a_{acc}$) represents the deceleration (acceleration) command with a step size of 3 m/s and a range within [0, 30] m/s. In the control module, we used the popular IDM model [29] to generate continuous speed control signals. A traditional DQN algorithm [30] is introduced to develop the longitudinal control scheme. Specifically, a fully connected 4-layer neural network is trained to learn the $Q$-funciton $Q(s, a)$. In DQN, $Q(s_t, a_t)$ is defined as the expected reward starting from $s_t$ at time $t$ that makes decision $a_t$ at time $t$ and thereafter follows the policy:

$$a_t^* = \arg\max_{a_t \in \mathbb{A}} Q(s_t, a_t)$$
$$Q = E_\pi \{R_t | s_t, a_t\} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t, a_t \right\} \quad (8)$$

TABLE I

PARAMETERS OF THE DQN MODEL

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Decay factor | 0.5 | Number of training steps | 15,000 |
| Learning rate | $2 \times 10^{-4}$ | Number of testing steps | 15,000 |
| Update rate | 50 | Minimum speed | 0 |
| Batch size | 64 | Maximum speed | 30 |
| Network layer | 2×64×64×3 | Weight $w_1$ | 0.4 |
| Activation function | ReLU | Weight $w_2$ | 1 |

where $\gamma \in (0, 1)$ is a forgetting factor and the reward function $r$ is designed as

$$r = w_1 r_1 + w_2 r_2$$
$$r_1 = \frac{v_{\text{ego}} - v_{\text{ego,min}}}{v_{\text{ego,max}} - v_{\text{ego,min}}}, \quad r_2 = \begin{cases} -1, & \text{collision} \\ 0, & \text{otherwise} \end{cases}, \quad (9)$$

where $w_1$, $w_2$ are weighting factors, and the speed range of the ego vehicle is set as $v_{\text{ego}} \in [v_{\text{ego,min}}, v_{\text{ego,max}}]$.

The Stable-Baselines3 algorithm developed by [30] is introduced to obtain a reliable DQN implementation, and all DQN parameters are listed in Table I. The model is capable of performing typical longitudinal following tasks after training and has collected 500 sets of data for subsequent explanatory analysis.

*2) Lane-Change Behavior Prediction Using Deep Learning:* We trained a lane-change (LC)/no-lane-change (No LC) prediction model on the HighD dataset [31] using a deep learning model. While constructing this model, we followed steps and rules as [3]. Firstly, we defined the labels for LC behavior.

TABLE II
PARAMETERS OF THE DEEP LEARNING MODEL

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Batch size | 32 | Learning rate | 0.01 |
| Network layer | 7×64×64×2 | Activation function | ReLU |
| Optimizer | SGD | Criterion | CrossEntropyLoss |

Specifically, for vehicle ID (58) (shown in Fig. 3 (a)), the lateral position significantly increases during the lane change process. This change starts from the red dot (indicating the start of the lane change) and continues to the green dot (indicating the completion of the lane change). For No LC behavior, we take vehicle ID (17) (shown in Fig. 3 (b)) as an example. Although this vehicle starts continuous lateral movement from the red dot, it did not change lanes. If this continuous lateral movement lasts for more than 20 frames (each frame lasts 40 milliseconds), the instantaneous data of this frame is extracted as no-lane-changing behavior data. For feature selection, we defined the feature vector $x = [v_0, v_1, v_2, v_3, d_1, d_2, d_3]$. As shown in Fig. 3 (c), $v_0$, $v_1$, $v_2$, and $v_3$ represent the speeds (unit: m/s) of HV, FVC, RVT, and FVT, respectively. $d_1$, $d_2$, and $d_3$ represent the longitudinal distances (unit: m) of HV relative to FVC, RVT, and FVT, respectively. Given the decision representation corresponding to the feature vector is $a \in \{a_{\mathrm{LC}}, a_{\mathrm{LH}}\}$, where $a_{\mathrm{LC}}$ ($a_{\mathrm{LH}}$) represents the LC/No LC decision. Specifically, a fully connected 4-layer neural network is trained to learn the lane change prediction model. All parameters are listed in Table II. The model's accuracy has reached above 95% after training, and a subset of data has been obtained for further analysis.

## B. Feature Attribution-Targeted CEs

This section will demonstrate how the SVCE method explains and diagnoses the data-driven autonomous driving model mentioned above.

*1) Feature Attribution-Targeted CEs for Autonomous Following Vehicle Model:* We are analyzing the first query sample presented in Fig. 4(a), where the query $x = [-6.37, 17.97, 23.93]$ results in the decision of acceleration (ACC). Initially, we employ Shapley values to estimate the contribution of each feature to the decision. As illustrated in the force plot, the difference between the predicted value of a sample and the base value $f(x^b)$ is determined by each feature's contribution. Positive impacts are represented as red arrows, while negative impacts are shown as blue arrows, with their magnitude proportional to the contribution. The sum of all contributions yields a probability value of 0.47 for the current ACC decision. We convert these contribution values into an importance measure (*Middle Column*), which guides the generation of the counterfactual sample. If $m = 1$, feature $v$ is the most crucial, and the counterfactual sample results in a decision of DEC. The SVCE states the explanation: "*The host vehicle's speed v=17.97 is the most important, contributing 0.094 to the current prediction. If changing its value from 17.97 to 15.13 would lead to a shift in the model's decision from ACC to DEC.*" For the counterfactual explanation when $m$=2: both the speed of the host vehicle and the relative distance are key factors. The model's decision changes from ACC to DEC when the speed decreases from 17.19 to 15.06 and the relative distance increases from 23.93 to 24.90. This suggests that decreasing the speed while increasing the distance can lead to a deceleration decision.

In the case of the second query instance $x = [-5.09, 16.71, 31.92]$, as shown in Fig. 4 (b), the most significant feature is $d$. The SVCE states: "*The distance d=31.92 is the most important, contributing 0.134 to the current prediction. If its value is altered from 31.92 to 25.38, the prediction will shift from DEC to ACC.*" This alteration of the feature and the subsequent shift in decision-making may lack logical consistency, as a decrease in distance typically should not be a reason for acceleration. These insights provide perspectives for understanding and diagnosing the model during its development. When $m = 2$, the counterfactual explanation considers both the relative speed and the relative distance. The relative speed increases from $-5.09$ to $-0.14$, and the relative distance decreases from 31.92 to 28.46, causing the model's decision to shift from DEC to ACC. This explanation holds logical coherence, as the rise in relative speed indicates that the leading vehicle is picking up pace relative to the host vehicle. This suggests the possibility of the leading vehicle accelerating. Consequently, if the gap between the host vehicle and the leading one is narrowing as well, the host vehicle's acceleration might be seen as an effort to prevent a substantial distance gap with the leading vehicle.

*2) Feature Attribution-Targeted CEs for LC Model:* Considering the first query sample shown in Fig. 5(a), the current decision is No LC. We aim to understand how each feature contributes to this decision to inspect potential biases and rationality in the model. Initially, we use Shapley values to quantify the contribution of each feature to the decision. We then transform these contribution values into an importance measure (*Middle Column*), guiding counterfactual sample construction. When $m = 1$, the most essential feature is $v_0$. We generate the current counterfactual sample and its corresponding decision: LC (*Right Column*). The SVCE states: *"The host vehicle's speed $v_0$=10.09 is the most important, contributing 0.28 to the current prediction. If its value increases from 10.09 to 19.699, the model's decision will change from No LC to LC"*. This explanation aligns with human understanding: When the vehicle's speed increases and the vehicle ahead is slower, it will perform a lane change to obtain higher speed, with $v_0 = 19.69$ as the decision boundary. When $m = 2$, the counterfactual explanation considers both the host vehicle's speed and the relative distance to the vehicle behind in the target lane. Specifically, when the host vehicle's speed increases from 10.09 to 13.76, and the relative distance to the following vehicle in the target lane increases from 11.92 to 15.97, the model's decision shifts from No LC to LC. The rationality of this explanation is reflected in several aspects: Firstly, the increase in the host vehicle's speed suggests a desire to maintain or enhance its current speed, and seeking a faster lane aligns with typical driver behavior. Secondly, the increase in relative distance provides a safer spatial condition for lane changing.

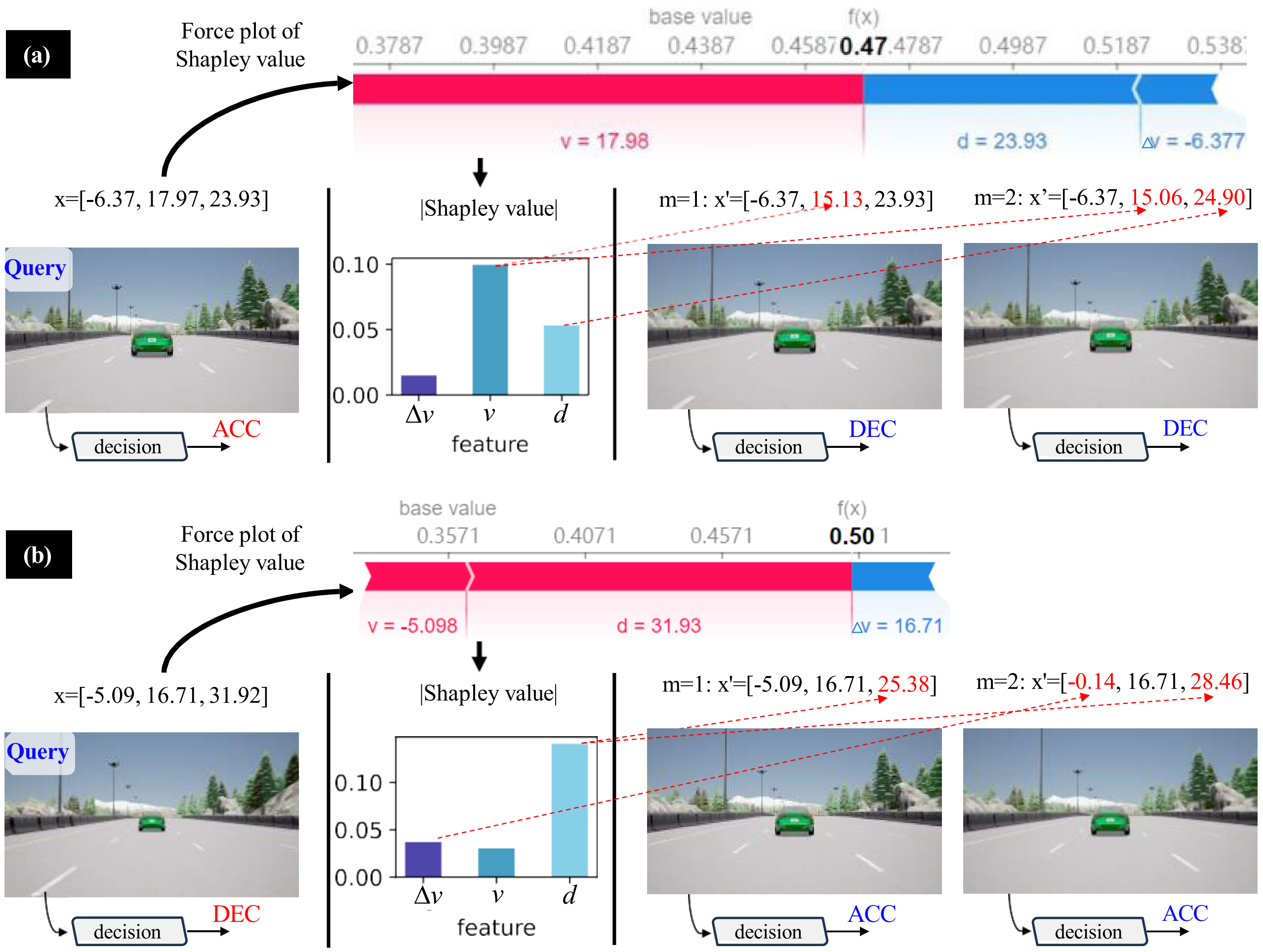


Fig. 4. Feature attribution-targeted counterfactual explanation for DQN decision-making in longitudinal following. For the queried samples and their associated decisions (*left column*), we employ shapley values (force plot of Shapley value is reflected as the reason for the difference between the predicted value $f(x)$ of a given sample and the base value $f(x^b)$, where the red arrows indicate that the feature contributes positively to the current decision, while blue arrows indicate a negative contribution, and the length of the arrows represents the magnitude of the contribution.) to generate feature importance (*middle column*) for each queried sample, then feature importance guides the construction of counterfactual samples associated with different decisions (*right column*) using the top $m$ important features.

For the second query instance $x = [23.90, 21.98, 34.68, 27.00, 21.07, 59.06, 93.22]$ shown in Fig. 5(b), the most significant feature is $d_2$ (*Right Column*). The SVCE states: *"The $d_2$=59.06 is the most important, contributing 0.28 to the current prediction. If its value decreases from 59.06 to 15.26, the model's decision will change from LC to No LC"*. This explanation intriguingly aligns with human intuition. Specifically, decreasing the relative distance between the host vehicle and the vehicle behind in the target lane could increase the risk of collision if it changes lanes. These observations offer viewpoints for comprehending the model. When $m = 2$, the counterfactual explanation considers both the distance to the vehicle behind in the target lane and the relative distance to the vehicle ahead in the target lane, with the distance to the vehicle behind decreasing from 59.06 to 35.43 and the relative distance to the vehicle ahead increasing from 93.22 to 130.5. As a result, the model's decision shifts from LC to No LC. This explanation appears contradictory: a decrease in the distance to the vehicle behind in the target lane suggests a reduced safety space for changing lanes, typically decreasing the probability of a LC, while an increase in the relative distance to the vehicle ahead in the target lane suggests an increased safety space, typically increasing the probability of an LC.

Furthermore, a complex case of lateral and longitudinal integration of autonomous driving decision-making is presented in the Appendix A.

### C. Quality of SVCE

In assessing the effectiveness of counterfactual explanations, sparsity and proximity are two critical metrics. Specifically, let $x$ be the original sample and $x'$ the corresponding counterfactual explanation sample. Then, sparsity can be represented as the number of features that have changed, mathematically expressed as:

$$Sparsity(x, x') = \sum_{i=1}^{n} \mathbf{1}_{x_i \neq x'_i}, \tag{10}$$

where $n$ is the total number of features, $x_i$ and $x_i'$ are the $i$th feature of the original sample and the counterfactual explanation sample, respectively. The function $\mathbf{1}x_i \neq x_i'$ is

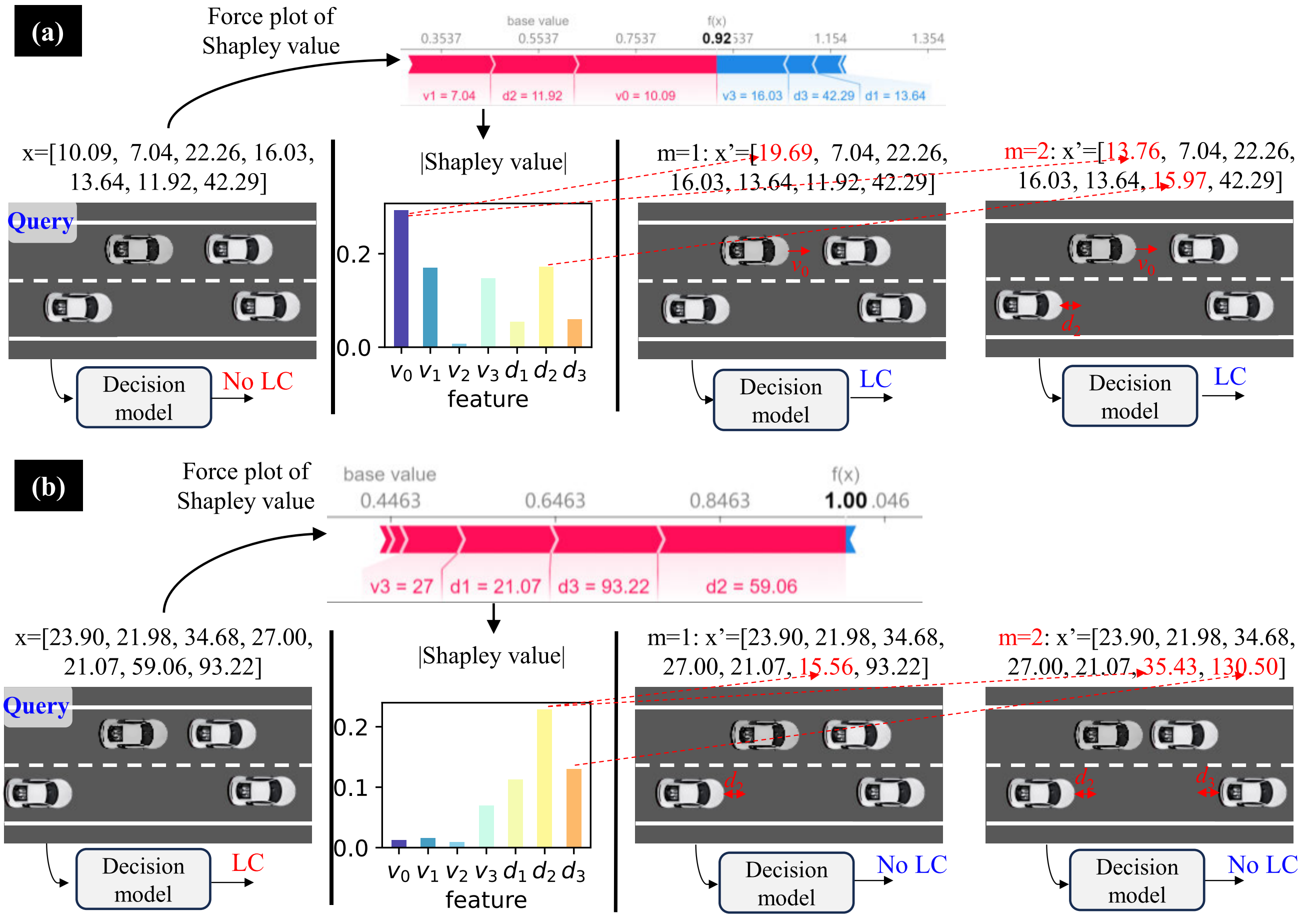

(a)
Force plot of Shapley value
base value
f(x)
0.3537
0.5537
0.7537
0.92
1.154
1.354
v1 = 7.04
d2 = 11.92
v0 = 10.09
v3 = 16.03
d3 = 42.29
d1 = 13.64
x=[10.09, 7.04, 22.26, 16.03, 13.64, 11.92, 42.29]
Query
Decision model
No LC
|Shapley value|
0.2
0.0
$v_0$ $v_1$ $v_2$ $v_3$ $d_1$ $d_2$ $d_3$
feature
m=1: x'=[19.69, 7.04, 22.26, 16.03, 13.64, 11.92, 42.29]
$v_0$
Decision model
LC
m=2: x'=[13.76, 7.04, 22.26, 16.03, 13.64, 15.97, 42.29]
$v_0$
$d_2$
Decision model
LC
(b)
Force plot of Shapley value
base value
f(x)
0.4463
0.6463
0.8463
1.00
v3 = 27
d1 = 21.07
d3 = 93.22
d2 = 59.06
x=[23.90, 21.98, 34.68, 27.00, 21.07, 59.06, 93.22]
Query
Decision model
LC
|Shapley value|
0.2
0.1
0.0
$v_0$ $v_1$ $v_2$ $v_3$ $d_1$ $d_2$ $d_3$
feature
m=1: x'=[23.90, 21.98, 34.68, 27.00, 21.07, 15.56, 93.22]
$d_2$
Decision model
No LC
m=2: x'=[23.90, 21.98, 34.68, 27.00, 21.07, 35.43, 130.50]
$d_2$
$d_3$
Decision model
No LC

Fig. 5. Feature attribution-targeted counterfactual explanation for deep learning-based lane-change model.

an indicator function that takes the value 1 when $x_i$ and $x'_i$ are different, and 0 otherwise. The proximity metric defines the closeness of the counterfactual instance to the original sample. For instance, using Euclidean distance, proximity can be defined as:

$$Proximity(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2}. \quad (11)$$

We randomly sampled 50 data each for lane-changing and longitudinal decision models, quantitatively comparing SVCE and Diverse Counterfactual Explanation (DiCE) [32]. As illustrated in Fig 6, SVCE demonstrates the better performance over the DiCE method in terms of both sparsity and proximity. This can be attributed to the utilization of Shapley values, which identify the features with the greatest impact on model decisions. Consequently, it achieves counterfactual explanations with fewer feature modifications and smaller changes in feature values.

### D. User Study

Given that the SVCE introduced in this paper are user-centric explanations, particularly within autonomous driving scenarios, it is essential to evaluate how these explanations assist users in understanding and diagnosing the model. Considering the above factors, we have chosen a user study to conduct our evaluation. As shown in Fig. 7, we designed a comparative experiment with a control group (not using SVCE explanation) and an SVCE group (using SVCE explanation), each consisting of 10 participants with basic knowledge of artificial intelligence. The survey is divided into two parts: the observation stage and the inquiry stage. In the observation stage, participants in the control group and the SVCE group each receive ten specific examples and the model's decision results in the given state. In addition, participants in the SVCE group will also receive SVCE counterfactual explanations for the corresponding examples. In the inquiry stage, all participants will receive two new examples, but the model's decision results for these examples will not be provided. They need to predict the model's decision on these new examples based on the knowledge gained in the observation stage.

*1) Consistency Ratio:* To quantitatively evaluate the effectiveness of the SVCE method in improving understanding of model decisions, we introduce the concept of prediction consistency ratio to measure the accuracy of participants' predictions. Let $P$ be the total number of predictions made by participants, and $C$ be the number of predictions that match the model's actual decisions, the consistency ratio is calculated as $R = \frac{C}{P}$. By comparing the consistency ratios of the control

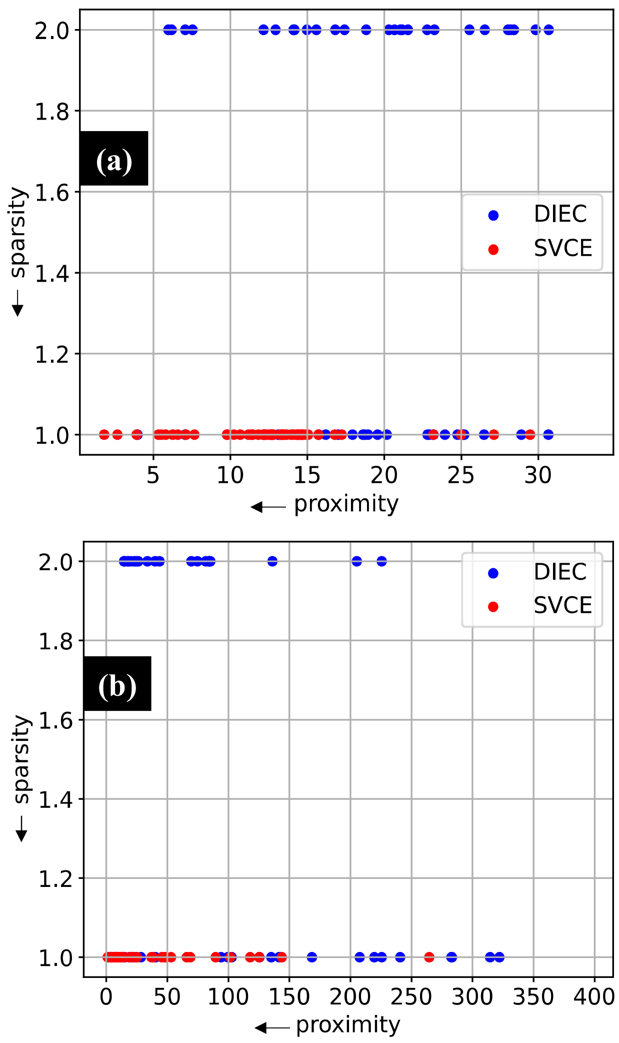
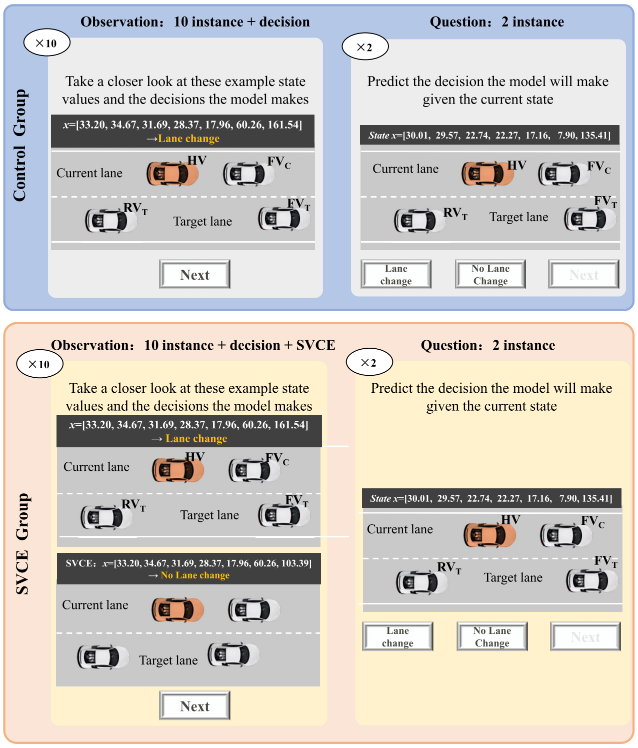


Fig. 6. **Comparison of sparsity and proximity** (Arrow notations ↓ and ← signify that lower metric values indicate better performance). (a) for the longitudinal following decision model, (b) for the LC decision model.

group and the experimental group, we can visually evaluate the impact of SVCE explanations on user understanding of model decisions. As shown in Table III, the experimental group achieved a prediction consistency ratio of 65%, which is higher than the control group's 40%.

*2) Bias Detection:* The consistency ratio measures the effectiveness of SVCE explanations in understanding the model. Another prospect of SVCE explanations is to enable bias detection. Thus, in the final section of the questionnaire, we specifically designed a survey with biased data. As illustrated in Fig. 10 (shown in appendix), if the leading vehicle in the target lane is close, the host vehicle is more inclined to change lanes, which reflects that the model has learned a bias inconsistent with normal driving cognition. The control group was still only given the original example without the SVCE explanation, while the SVCE group was provided with an explanation. Afterward, in the test samples, participants were asked to predict the model's decision and the reasons for their prediction based on observation. The reasons for decisions made by the respondents were analyzed to determine whether they identified any bias in the model.


Control Group

Observation：10 instance + decision

×10

Take a closer look at these example state values and the decisions the model makes

x=[33.20, 34.67, 31.69, 28.37, 17.96, 60.26, 161.54] →Lane change

Question：2 instance

×2

Predict the decision the model will make given the current state

State x=[30.01, 29.57, 22.74, 22.27, 17.16, 7.90, 135.41]

SVCE Group

Observation：10 instance + decision + SVCE

×10

Take a closer look at these example state values and the decisions the model makes

x=[33.20, 34.67, 31.69, 28.37, 17.96, 60.26, 161.54] → Lane change

SVCE：x=[33.20, 34.67, 31.69, 28.37, 17.96, 60.26, 103.39] → No Lane change

Question：2 instance

×2

Predict the decision the model will make given the current state

State x=[30.01, 29.57, 22.74, 22.27, 17.16, 7.90, 135.41]

Fig. 7. **Survey Design:** The survey consists of two stages. In the observation stage, participants will observe a series of examples and the model's predictions. In the questionnaire stage, participants will guess the model's predictions for new examples based on their observations in the observation stage. For the control group, they can only see the examples and their corresponding decision results in the observation stage. For the SVCE group, they can also receive additional SVCE information.

TABLE III

RESULTS OF USER STUDY

| | Control group | SVCE group |
|---|---|---|
| Consistency ratio | 40% | 65% |
| Bias detection | 0% | 60% |

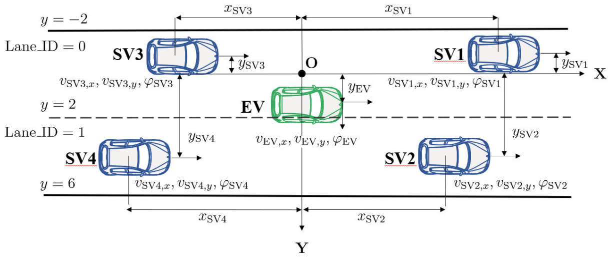

$x_{\mathrm{SV3}}$
$x_{\mathrm{SV1}}$
$y=-2$
Lane_ID = 0
SV3
$y_{\mathrm{SV3}}$
O
SV1
$y_{\mathrm{SV1}}$
X
$v_{\mathrm{SV3},x}, v_{\mathrm{SV3},y}, \varphi_{\mathrm{SV3}}$
$y_{\mathrm{EV}}$
$v_{\mathrm{SV1},x}, v_{\mathrm{SV1},y}, \varphi_{\mathrm{SV1}}$
$y=2$
EV
Lane_ID = 1
$y_{\mathrm{SV4}}$
$v_{\mathrm{EV},x}, v_{\mathrm{EV},y}, \varphi_{\mathrm{EV}}$
$y_{\mathrm{SV2}}$
SV4
SV2
$y=6$
$v_{\mathrm{SV4},x}, v_{\mathrm{SV4},y}, \varphi_{\mathrm{SV4}}$
$v_{\mathrm{SV2},x}, v_{\mathrm{SV2},y}, \varphi_{\mathrm{SV2}}$
$x_{\mathrm{SV4}}$
$x_{\mathrm{SV2}}$
Y

Fig. 8. **State space diagram** [33].

In Table III, we observed that 60% of the participants in the experimental group identified the model's unreasonable correlations and made the same predictions as the model. In contrast, no one in the control group mentioned this. This result clearly demonstrates the effectiveness of the SVCE approach in identifying bias within the model's decision-making. Detailed survey results are presented in Table IV (shown in the appendix).

TABLE IV

SURVEY RESULT ('–' DENOTES AN INVALID EXPLANATION, MARK EXPLANATIONS THAT IDENTIFY BIAS IN RED FONT, THE MODELS DECISION IS NO LANE CHANGE)

| Group | Research Background | Prediction | Please explain the factors that drove your choice |
|---|---|---|---|
| Control Group | Autonomous driving decision-making optimization based on reinforcement learning | Lane change × | The safety of changing lanes is proportional to the distance from the car ahead on the target lane |
| | Research and implementation of human-like autonomous driving decision algorithms | Lane change × | – |
| | Intelligent vehicle model predictive control methods | Lane change × | – |
| | Data-driven vehicle parameter estimation techniques | Lane change × | A greater distance from the vehicle ahead means less pressure during lane changing |
| | Testing methods for the intelligence level of autonomous vehicles | Lane change × | Its advisable to change lanes when theres significant space from the vehicle ahead in the intended lane |
| | Research on intelligent trajectory planning algorithms under vehicle-road coordination | Lane change × | Lane transitioning is safer when the lead vehicle on the desired lane is further away |
| | Commercial vehicle energy-saving optimization | Lane change × | The wider the space on the next lane, the lower the risk when changing lanes |
| | Decision-making for exiting ramp in dense traffic environment | Lane change × | – |
| | Autonomous driving decision system based on deep learning | Lane change × | When changing lanes, a farther leading vehicle on the target lane allows for a more relaxed maneuver |
| | Vehicle dynamic characteristics modeling and simulation | Lane change × | The increase in distance ahead provides a better opportunity for lane changing |
| SVCE Group | Model predictive control | No Lane change ✓ | In the observation, the further the distance to the leading vehicle in the target lane, the more the model tends to make a decision not to change lanes |
| | Vehicle behavior prediction and trajectory planning | No Lane change ✓ | – |
| | Multi-vehicle collaborative decision-making planning | No Lane change ✓ | The likelihood of the model choosing not to switch lanes rises with the distance to the vehicle in front on the target lane |
| | Human-machine collaborative decision control | No Lane change ✓ | The greater the distance to the vehicle ahead in the target lane, the less likely the model is to opt for a lane change |
| | Design and optimization of human-machine interface for autonomous vehicles | No Lane change ✓ | The leading vehicle in the current lane is obstructing the host vehicle |
| | Decision-making planning based on reinforcement learning | No Lane change ✓ | – |
| | Autonomous vehicle decision planning control | No Lane change ✓ | The probability of the model deciding to stay in its lane increases as the gap to the leading vehicle in the target lane widens |
| | Autonomous driving and decision planning | No Lane change ✓ | As the space between the model and the leading vehicle in the target lane expands, the model's inclination to keep its lane intensifies |
| | Commercial vehicle energy-saving optimization control | No Lane change ✓ | – |
| | End-to-end autonomous driving decision-making | No Lane change ✓ | During the observation phase, the further the vehicle ahead in the target lane, the less inclined the host vehicle is to change lanes |

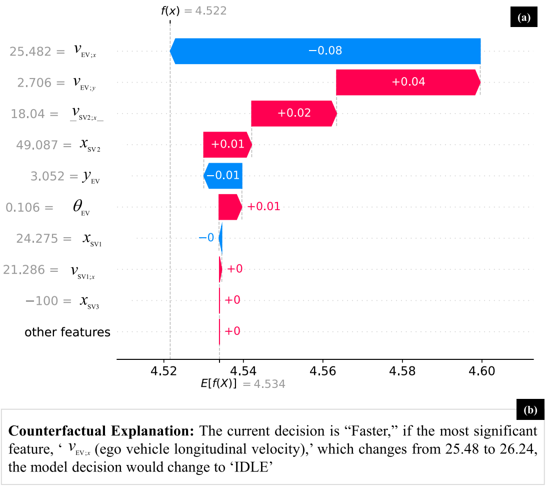
## V. CONCLUSION

In this study, we presented SVCE, targeting scenarios in autonomous driving. The fundamental idea of this approach is to utilize the Shapley value-based feature contribution to direct the counterfactual generation model to concentrate on the most impactful input features. This strategy allows end users to swiftly identify the search direction for generating CEs, such as prioritizing the most crucial features and producing representative CEs. In contrast to previous studies, our method autonomously generates CEs, eliminating manual feature selection requirements. We evaluated our method using two data-driven autonomous driving decision and behavior prediction models. The results confirm the effectiveness of our approach in providing insightful explanations, identifying decision boundaries, and uncovering model biases. Furthermore, we conducted a user-centric study to assess the practical applicability of SVCE in autonomous driving scenarios, which serves as a crucial validation of the presented SVCE.

While SVCE currently focuses on generating CEs for structured data, we recognize the critical role that visual inputs play in autonomous driving models. Given the recent rapid progress in image generation models, integrating this technology with SVCE could potentially emerge as an interesting direction for future research.

## APPENDIX A
## EXPLANATION FOR INTEGRATED LONGITUDINAL AND LATERAL DECISION MODEL

To demonstrate the application of the proposed method in more complex driving environments, we have established an integrated longitudinal and lateral decision model for typical highway scenarios Fig 8 illustrates the schematic of the state



**Counterfactual Explanation:** The current decision is "Faster," if the most significant feature, ' $v_{EV;x}$ (ego vehicle longitudinal velocity),' which changes from 25.48 to 26.24, the model decision would change to 'IDLE'

Fig. 9. **SVCE Explanations for the lateral and longitudinal decision–making model.**(a) Shapley values of the features, and (b) counterfactual explanations.

**Algorithm 1** Neural Network-Specific SVCE

```
Input: instance $x$, model $f$, baseline $x^b$, $m$, $M$
Output: $x'$, $flag$
Initialize successful flag $flag \leftarrow$ False
Initialize counterfactual example $x' \leftarrow x$
$\phi \leftarrow$ Calculate Shapley $(model, x, x^b)$ on (3)
$\phi' \leftarrow |\phi|$ on (5)
for step = 1, $M$ do
    Perform a gradient descent on (7) with respect to the $x'$
    if $f(x) \neq f(x')$ then
        $flag \leftarrow$ True
        break
    end if
end for
return $x'$, $flag$
```

**Algorithm 2** Model-Agnostic SVCE

```
Input: instance $x$, model $f$, baseline $x^b$, $m$, $M$, $n$, $\epsilon$
Output: $x'$, $flag$
Initialize successful flag $flag \leftarrow$ False
Initialize counterfactual example $x' \leftarrow x$
Initialize $I \leftarrow \arg\max_{I} \left(\{\phi'_i\}_{i=1}^{n}, m\right)$
$\phi \leftarrow$ Calculate Shapley $(model, x, x^b)$ on (3)
$\phi' \leftarrow |\phi|$ on (5)
$x'^{,f} \leftarrow x'$
$x'^{,b} \leftarrow x'$
for step = 1, $M$ do
    for $i$ = 1, $n$ do
        if $i \in I$ then
            $x'^{,f} \leftarrow x'^{,f} + \epsilon$
            $x'^{,b} \leftarrow x'^{,b} - \epsilon$
        end if
    end for
    if $f(x) \neq f(x'^{,f})$ then
        $x' \leftarrow x'^{,f}$
        $flag \leftarrow$ True
        break
    else if $f(x) \neq f(x'^{,b})$ then
        $x' \leftarrow x'^{,b}$
        $flag \leftarrow$ True
        break
    end if
end for
return $x'$, $flag$
```

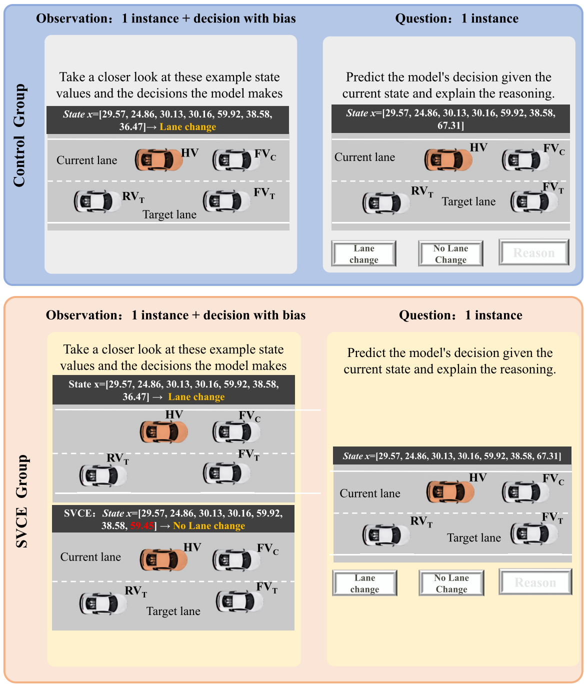

Control Group
Observation: 1 instance + decision with bias
Take a closer look at these example state values and the decisions the model makes
State x=[29.57, 24.86, 30.13, 30.16, 59.92, 38.58, 36.47]→ Lane change
Current lane
HV
$FV_C$
$RV_T$
$FV_T$
Target lane
Question: 1 instance
Predict the model's decision given the current state and explain the reasoning.
State x=[29.57, 24.86, 30.13, 30.16, 59.92, 38.58, 67.31]
Current lane
HV
$FV_C$
$RV_T$
Target lane
$FV_T$
Lane change
No Lane Change
Reason
SVCE Group
Observation: 1 instance + decision with bias
Take a closer look at these example state values and the decisions the model makes
State x=[29.57, 24.86, 30.13, 30.16, 59.92, 38.58, 36.47] → Lane change
HV
$FV_C$
$RV_T$
$FV_T$
SVCE: State x=[29.57, 24.86, 30.13, 30.16, 59.92, 38.58, 59.45] → No Lane change
Current lane
HV
$FV_C$
$RV_T$
Target lane
$FV_T$
Question: 1 instance
Predict the model's decision given the current state and explain the reasoning.
State x=[29.57, 24.86, 30.13, 30.16, 59.92, 38.58, 67.31]
Current lane
HV
$FV_C$
$RV_T$
Target lane
$FV_T$
Lane change
No Lane Change
Reason

Fig. 10. **Bias detection:** Provide an example with bias, where the control group is not provided with counterfactual explanations, and the SVCE group is provided with counterfactual explanations. Then, use a new example to ask participants to predict the outcome and provide reasons for their prediction.

space, including an Ego Vehicle (EV) and several Surrounding Vehicles (SVs). The SVs execute given high-level decisions such as lane changes and acceleration/deceleration using the Intelligent Driver Model (IDM) [29] for longitudinal control and the MOBILE [34] model for lateral control. Four surrounding vehicles (SVs) are considered as traffic participants influencing the decision of the EV, located as follows: one in front of the EV, one behind the EV, one in front of the EV on another lane, and one behind the EV on another lane. For each surrounding vehicle, the considered states include: longitudinal position $x_{\mathrm{SV}i}$, lateral position $y_{\mathrm{SV}i}$, longitudinal velocity $v_{\mathrm{SV}i;x}$, lateral velocity $v_{\mathrm{SV}i;y}$, and orientation angle $\theta_{\mathrm{SV}i}$, where $i = 1, \ldots, 4$. The longitudinal relative position of the EV is always fixed at $x_{\mathrm{EV}} = 0$ meters, while its lateral position is represented by $y_{\mathrm{EV}}$. The longitudinal and lateral velocities of the EV, as well as its orientation angle, are denoted by $v_{\mathrm{EV};x}$, $v_{\mathrm{EV};y}$ and $\theta_{\mathrm{EV}}$, respectively. The reward function adopts the same setup as the longitudinal following model, as described in (9). The model parameters are consistent with the DQN settings as Tabel I, with the number of training steps set to 100,000. Due to changes in the dimensions of the action and state spaces, the network architecture is adjusted to 25*64**64*5.

As shown in Fig. 9 (a), the ego vehicle speed has the most significant impact (most considerable absolute value) on the current 'Faster' decision. Using the SVCE method to modify this feature, the counterfactual explanation in Fig. 9 (b) shows that if the ego vehicle speed increased from 25.48 to 26.24, the model decision would change from 'Faster' to 'IDLE'. This explanation reflects that if the current vehicle speed were to increase, the model would not accelerate to ensure driving safety.

## Appendix B
## Pseudocode for SVCE Implementation

See Algorithm 1.

## Appendix C
## Survey

See Algorithm 2, Fig. 10, and Table IV.

## References

[1] P. R. Wurman et al., "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, Feb. 2022.

[2] S. Feng et al., "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, Mar. 2023.

[3] M. Li, Y. Wang, H. Sun, Z. Cui, Y. Huang, and H. Chen, "Explaining a machine-learning lane change model with maximum entropy Shapley values," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 6, pp. 3620–3628, Jun. 2023.

[4] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, "A survey on trajectory-prediction methods for autonomous driving," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 652–674, Sep. 2022.

[5] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowl.-Based Syst.*, vol. 214, Feb. 2021, Art. no. 106685.

[6] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard J. Law Technol.*, vol. 31, no. 2, p. 841, 2017.

[7] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: Challenges revisited," 2021, *arXiv:2106.07756*.

[8] J. Kim and J. Canny, "Interpretable learning for self-driving cars by visualizing causal attention," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2942–2950.

[9] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze, "Explanations in autonomous driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10142–10162, Aug. 2022.

[10] M. Ivanovs, R. Kadikis, and K. Ozols, "Perturbation-based methods for explaining deep neural networks: A survey," *Pattern Recognit. Lett.*, vol. 150, pp. 228–234, Oct. 2021.

[11] A. Samadi, A. Shirian, K. Koufos, K. Debattista, and M. Dianati, "SAFE: Saliency-aware counterfactual explanations for DNN-based automated driving systems," 2023, *arXiv:2307.15786*.

[12] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: Deep learning for interpretable image recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–11.

[13] Q. Zhang, Y. N. Wu, and S.-C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8827–8836.

[14] Y. Shen, S. Jiang, Y. Chen, and K. Driggs Campbell, "To explain or not to explain: A study on the necessity of explanations for autonomous vehicles," 2020, *arXiv:2006.11684*.

[15] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (XAI): Toward medical XAI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, Aug. 2020.

[16] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *Int. J. Comput. Vis.*, vol. 130, pp. 2425–2452, Aug. 2022.

[17] Q. Zhang, X. J. Yang, and L. P. Robert, "Expectations and trust in automated vehicles," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, Apr. 2020, pp. 1–9.

[18] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[20] D. Fryer, I. Strumke, and H. Nguyen, "Shapley values for feature selection: The good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144352–144360, 2021.

[21] T. Heskes, E. Sijben, I. G. Bucur, and T. Claassen, "Causal Shapley values: Exploiting causal knowledge to explain individual predictions of complex models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 4778–4789.

[22] L. He, N. Aouf, and B. Song, "Explainable deep reinforcement learning for UAV autonomous path planning," *Aerosp. Sci. Technol.*, vol. 118, Nov. 2021, Art. no. 107052.

[23] S. Tang et al., "Data valuation for medical imaging using Shapley value and application to a large-scale chest X-ray dataset," *Sci. Rep.*, vol. 11, no. 1, p. 8366, Apr. 2021.

[24] A. Lahiri, K. Alipour, E. Adeli, and B. Salimi, "Combining counterfactuals with Shapley values to explain image models," 2022, *arXiv:2206.07087*.

[25] Y. Ramon, D. Martens, F. Provost, and T. Evgeniou, "A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C," *Adv. Data Anal. Classification*, vol. 14, pp. 801–819, Dec. 2020.

[26] E. Algaba, V. Fragnelli, and J. Sánchez-Soriano, *Handbook Shapley Value*. Boca Raton, FL, USA: CRC Press, 2019.

[27] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–4.

[28] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, vol. 78, 2017, pp. 1–16.

[29] M. Zhou, X. Qu, and S. Jin, "On the impact of cooperative autonomous vehicles in improving freeway merging: A modified intelligent driver model-based approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1422–1428, Jun. 2017.

[30] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *J. Mach. Learn. Res.*, vol. 22, no. 268, pp. 12348–12355, 2021.

[31] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.

[32] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proc. Conf. Fairness, Accountability, Transparency*, Jan. 2020, pp. 607–617.

[33] Z. Cui, Y. Wang, N. Bian, and H. Chen, "Reward machine reinforcement learning for autonomous highway driving: An unified framework for safety and performance," in *Proc. 7th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2023, pp. 1–6.

[34] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 86–94, Jan. 2007.

**Meng Li** received the B.S. degree from the College of Electronic and Information Engineering, Changchun University of Science and Technology, China, in 2017, and the M.S. degree in control science and engineering from Jilin University, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Control Science and Engineering, Tongji University, China. His research interests include intelligent driving, artificial intelligence, explainable AI, and autonomous vehicle.

**Hengyang Sun** received the B.S. degree from the School of Automotive Studies, Tongji University, China, in 2023, where he is currently pursuing the M.S. degree. His research interests include machine learning, deep learning, explainable AI, and autonomous vehicle.

**Yanjun Huang** received the Ph.D. degree from the Department of MME, University of Waterloo, in 2016. He is currently a Professor with the School of Automotive Studies, Tongji University. He has published several books, over 60 papers in journals and conferences. His research interests include improving vehicle performance in terms of safety, energy-saving, and intelligence by using advanced control and learning methods. He was a recipient of the IEEE VTS 2019 Best Land Transportation Paper Award and the 2018 Best Paper of Automotive Innovation. He is serving as an AE or EBM for *IET Intelligent Transport Systems*, *SAE International Journal of Commercial Vehicles*, and *International Journal of Autonomous Vehicle Systems*.

**Hong Chen** (Fellow, IEEE) received the B.S. and M.S. degrees in process control from Zhejiang University, China, in 1983 and 1986, respectively, and the Ph.D. degree in system dynamics and control engineering from the University of Stuttgart, Germany, in 1997. In 1986, she joined Jilin University of Technology, China. From 1993 to 1997, she was a Wissenschaftlicher Mitarbeiter with Institut fuer Systemdynamik und Regelungstechnik, University of Stuttgart. Since 1999, she has been a Professor with Jilin University and hereafter a Tang Aoqing Professor. She is currently a Distinguished Professor with Tongji University. Her current research interests include model predictive control, nonlinear control, artificial intelligence, and applications in mechatronic systems (automotive systems).