Expected Integral Discrete Gradient: Diagnosing Autonomous Driving Model

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

***Abstract*—The demand for explainability in complex machine learning (ML) models is ever more pressing, particularly within safety-critical domains like autonomous driving. The Integrated Gradient (IG), a prominent attribution-based explainable artificial intelligence method, offers an effective solution for explaining and diagnosing ML models. However, IG is primarily designed for deep neural network models, hindering its broader applicability to various structured machine-learning models. Moreover, the issue of selecting a suitable single baseline point in IG methods still needs to be solved. In response to these challenges, this paper introduces a model-agnostic explainable technique called the Expected Integral Discrete Gradient (EIDG). This approach extends the capabilities of IG to encompass a wide range of machine-learning models by leveraging numerical differentiation. It replaces the previous single baseline point scheme with a distributed multi-baseline method to reveal how varying baselines affect the output. Our method is thoroughly evaluated on standard machine learning models, targeting scenarios in autonomous driving scenarios, thereby validating its effectiveness in explaining and diagnosing models. Our work will inspire and equip developers and users with the necessary tools to promote the adoption of attribution explanations across various machine-learning domains. The code for EIDG is available at https://github.com/limeng-1234/EIDG.**

***Index Terms*—Explainable artificial intelligence, Integrated gradient, Machine learning, Autonomous driving.**

I. Introduction

MACHINE learning models are increasingly being employed, particularly in safety-critical applications such as wireless communications [1], [2] and autonomous driving [3]–[6]. Due to the inherent black-box nature of machine learning , there is a rising demand for model explainability [7]–[10]. This demand is reflected in calls for explanations by diverse regulatory bodies, such as the General Data Protection Regulation's 'right to explanation.' [11]. Thus, explanations are essential for legal liability and enhancing end-user trust, especially targeting autonomous driving [12]–[15].

This work was supported in part by the National Natural Science Foundation of China, in part by the Joint Fund for Innovative Enterprise Development under Grant U23B2061, in part by the Fundamental Research Funds for the Central Universities, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, and in part by Xiaomi Young Talents Program/Xiaomi Foundation. (Corresponding author: Yanjun Huang. Authors Meng Li and Hengyang Sun contributed equally to this article)

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

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

Numerous methods are available for explaining machine learning models. Among them, the local feature attribution approach has gained substantial popularity [16]. Formally, within this kind of approach, given a model $f : \mathbb{R}^n \to \mathbb{R}$, for an n-dimensional input x, feature attribution assigns a contribution value $atr_i(x; f) \in \mathbb{R}$ to each feature $i \in [1 \cdots d]$. Therefore, it could be utilized within a machine learning-driven field to explain the prediction [17], [18].

While multiple very good frameworks are dedicated to attribution explanations [19]–[22], the currently superior architecture is Integrated Gradients (IG) [23], which satisfies both *completeness* and *invariance*. The fundamental concept of IG is to evaluate feature contributions by averaging the gradients of the model's output along linear paths from a baseline to a prediction point. Therefore, we decided to use IG as a starting point for our research. By analyzing the IG, we will improve our work in two aspects: First, it is worth noting that the IG method is constrained to deep neural network models, limiting its use in various structured machine-learning models. Our first main idea is, therefore, to employ numerical differentiation and expand the scope to different machine-learning models; Second, we believe that utilizing a single baseline point within the IG framework is not a good architecture choice. Recent literature [20], [21] has demonstrated that selecting a singular baseline point often yields biased attribution outcomes. Consequently, it is proposed to model attribution by calculating the expected attribution outcomes across various baseline points. Our second main idea is to propose a multi-baseline scheme for the IG framework and formalize the baseline design.

Empirically, we demonstrate the effectiveness of our proposed approach by evaluating it on diverse structured machine-learning models, focusing on scenarios in autonomous driving. In summary, The main contributions are threefold:

- We employ numerical differentiation methods to extend the previously deep learning-specific IG explanation to any machine learning model.
- And we comprehensively analyze the proposed EIDG employing the autonomous driving models. The results provide empirical evidence for insightful model explanations and discovering model biases.
- We offer a model-agnostic attribution explanation toolkit, which allows developers and users to be inspired and equipped with the necessary tools for promoting the adoption of attribution methods across different machine-learning domains.

The remainder of this paper is organized as follows. Section III describes the proposed Expected Integral Discrete Gradient

(EIDG) method. Section IV presents the evaluation results and discusses their implications. Finally, we conclude this work in Section V.

II. Background and Related Works

Local Feature Attribution Strategy. Local feature attribution calculates importance scores or contributions of each feature to the model's prediction and has been extensively studied [19]–[22].

One category of methods is model-agnostic, indicating their independence from specific model architectures. A prominent technique within this category is Local Interpretable Model-agnostic Explanations (LIME) [19]. LIME generates a set of samples near an input and trains an explanatory model (typically a linear model or decision tree) with these samples to capture the behavior of the original model. The weights of this model represent the feature contributions. The Shapley value algorithm [21] considers the marginal contributions of features across all possible combinations and models attribution as the average change in output under different combinations.

Another category consists of methods specific to model structures, particularly tailored for neural network architectures. DeepLIFT [20] quantifies the contributions of input features relative to a reference value (typically the feature's mean or median). Through decomposition rules, DeepLIFT precisely delineates how each input feature's dependency on the model output is distributed across the network's layers. DeepSHAP [21], based on the principles of Shapley values and DeepLIFT integrates the game-theoretic Shapley values method with deep learning's backpropagation techniques. SHAP values are calculated based on a fair distribution principle, where each feature is considered a "player" whose contribution is determined by simulating how the model output would change without that feature. By leveraging DeepLIFT's backpropagation, DeepSHAP efficiently assigns a score to each input feature in deep networks, clearly indicating the positive or negative impact of these features on model predictions. Layer-wise Relevance Propagation (LRP) [22] identifies the most significant contributing input features by propagating the prediction differences back through the layers from the output. Starting from the model's last layer, LRP forwards each node's output contribution (relevance scores) to the input layer. This layered relevance propagation mechanism allows LRP to reveal how each node within the network impacts the final decision-making process. Regrettably, the attribution above methods do not simultaneously satisfy the axioms of *completeness* and *invariance* [23], highlighting significant limitations of these techniques. Guided by these hypotheses, Integrated Gradients (IG) [23] was introduced. The fundamental concept of IG is to assess feature contributions by averaging the gradients of the model's output along a linear path from the baseline to the prediction point. However, it is important to note that the IG method is restricted to deep neural network models, limiting its use across various structured machine learning models. The first part of our work builds upon the foundation of IG, extending the IG method to any machine-learning model using a discrete gradient approach.

Baselines selection. The baseline or reference value plays a crucial role in feature attribution methods. It serves as the reference point for comparing and evaluating the impact of each input feature on the model's output. By defining a baseline, we can quantify the changes in model predictions when input features move from the baseline to their actual observed values. This comparison reveals the contributions of each feature to the model's output, thereby aiding our understanding of the decision-making process within the model. The choice of baseline significantly affects the results of model explanations, making it an essential step in feature attribution analysis. Definitions of the baseline can vary significantly across different methods and applications. In DeepLIFT, the baseline is typically chosen as the mean or median of the feature values in the training data, assuming that the model's output under these baseline conditions represents the performance in these average or median states. When dealing with image or text data, the baseline might be set to all zeros, helping us understand how transitioning from a "no information" state to an "information" state affects model decisions. Recent literature [20], [21] suggests that choosing a single baseline point produces biased attribution results. Inspired by this, Our second major idea is to adopt a multiple baseline scheme for the IG framework and formalize the baseline design.

III. Expected Integral Discrete Gradient

This section outlines the EIDG method, which consists of two primary components. The initial component extends the applicability of integrated gradients to a broader range of machine-learning models by employing numerical differentiation, as discussed in Section III-A. The second component introduces a multi-baseline scheme, as explained in Section III-B. Subsequently, Section III-C provides a detailed account of the explainability axioms adhered to by EIDG. Fig. 1 offers an overview of the proposed EIDG method. In Fig. 1(a), the fundamental concept of IG is to evaluate feature contributions by averaging the gradients of the models output along linear paths from a baseline x' to a prediction point x, which is designed for the neural network. As shown in Fig. 1(b), EIDG extends the IG to arbitrary models using numerical differentiation methods and adopts a multi-baseline approach to overcome potential attribution biases caused by single baselines. Fig. 1(c) demonstrates how EIDG provides attribution explanations in a data-driven autonomous driving model.

A. Integral Discrete Gradient

Considering a model f with input $x \in \mathbb{X}$ and reference input x', the IG method computes the attribution of the i-th feature by integrating the gradient along a straight path from the baseline x' to the input x multiplied by $x_i - x'_i$ [23]:

$$\begin{aligned} \mathrm{IG}_i(x) &= (x_i - x'_i) \\ &\times \Sigma_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}, \end{aligned} \tag{1}$$

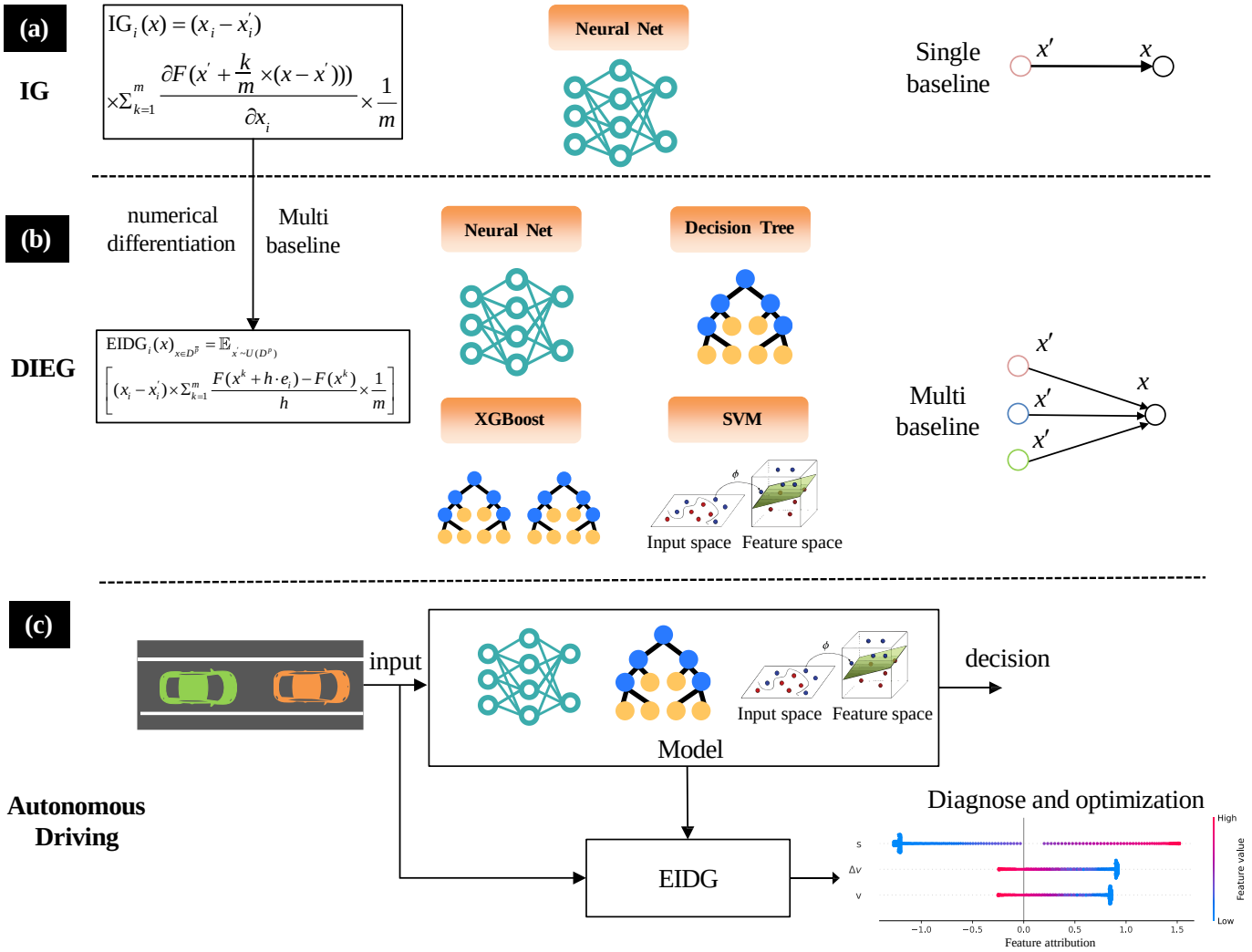


Fig. 1: **Brief overview of this work**. **a**. The fundamental concept of IG is to evaluate feature contributions by averaging the gradients of the models output along linear paths from a baseline x' to a prediction point x, which is designed for neural network. **b**. EIDG extends the IG to arbitrary models using numerical differentiation methods and adopts a multi-baseline approach to overcome potential attribution biases caused by single baselines. **c**. In the context of autonomous driving applications, employing EIDG to obtain attribution values for all features and visualizing them facilitates the diagnosis and optimization of the model.

where m is the total number of steps considered for the approximation. However, the IG attribution method, as represented by (1), is currently limited to application only within neural network models. Consequently, we resort to forward numerical differentiation methods to approximate gradients, extending the IG method to arbitrary models:

$$\frac{\partial F(x^k)}{\partial x_i} \approx \frac{F(x^k + h \cdot e_i) - F(x^k)}{h}, \qquad x^k = \frac{k}{m} \times (x - x') \tag{2}$$

where e_i represents a vector of the same dimension as x^k, where the i-th element is 1, and all other elements are 0. h represents the increment and is a candidate parameter. Combining (1) and (2), we obtain the integrated discrete gradient:

$$\mathrm{IG}_i(x) = (x_i - x_i') \times \Sigma_{k=1}^{m} \frac{F(x^k + h \cdot e_i) - F(x^k)}{h} \times \frac{1}{m}. \tag{3}$$

B. Expected Baseline with Positivity

The choice of baseline significantly affects the attribution, making baseline selection an open problem. Therefore, this section elucidates the baseline design of EIDG.

1) Expected Baseline: Inspired by [20], [21], EIDG employs a multiple baseline strategy to address this challenge, offering several advantages. It captures diverse perspectives on the issue by calculating an expected baseline, revealing how different input values affect the output in various ways. Furthermore, the multiple baselines are less susceptible to the

influence of noise or outliers than a single baseline, leading to enhanced stability in explanations. The core idea of multiple baselines for EIDG is to randomly sample multiple baselines from the data set D and perform an expected operation on the IG values obtained for each baseline:

$$\mathrm{IG}_i(x) = \mathbb{E}_{x' \sim U(D)} \left[(x_i - x_i') \times \Sigma_{k=1}^{m} \frac{F(x^k + h \cdot e_i) - F(x^k)}{h} \times \frac{1}{m} \right]. \tag{4}$$

Considering the completeness axiom of IG, $F(x) = \Sigma_{i=1}^{n} \mathrm{IG}_i(x) + F(x')$, where n is the feature number of x, we can directly derive the completeness axiom for EIDC:

$$F(x) = \Sigma_{i=1}^{n} \mathrm{IG}_i(x) + \mathbb{E}_{x' \sim U(D)}[F(x')]. \tag{5}$$

Generally, for a decision model, using subscript a to represent the different prediction channels of the model, (5) can be expressed as:

$$F_a(x) = \Sigma_{i=1}^{n} \mathrm{IG}_i(x, a) + \mathbb{E}_{x' \sim U(D)}[F_a(x')]. \tag{6}$$

2) Positive Expected Baseline: We further establish the principles for selecting the baseline through an axiomatized description. Intuitively, in a decision model F with input x, for the selected decision a^*, it can be considered that all the features together contribute to the success of that decision. Therefore, the sum of the contributions of all features to the chosen decision should be positive. We define this property as *positivity*, denoted as

$$\sum_{i=1}^{n} \mathrm{IG}_i\left(x, a^*\right) \geq 0 \tag{7}$$

To ensure that the sum of attribution values satisfies (7), taking single baseline as an example, the maximum-minimum optimization method is employed to obtain the baseline:

$$x' = \arg \min_{x \in \mathbb{X}} \max_{a \in \mathbb{A}} F_a(x) \tag{8}$$

The baseline x' computed by (8) satisfies positivity. Please refer to the proof.

For multiple baselines, with p representing the number of baselines, (8) is computed iteratively p times to obtain the set D^p consisting of p single positive definite baselines. It can be easily demonstrated that for all x ($x \in D^{\bar{p}} = D - D^p$), the sum of feature attributions of the selected decision a^* will always be greater than or equal to zero:

$$\sum_{i=1}^{n} \mathrm{IG}_i(x, a^*)_{x \in D^{\bar{p}}} = F_{a^*}(x) - E_{x' \sim U(D^p)}[F_{a^*}(x')] \geq 0. \tag{9}$$

Then, combining (4) and (9), we can express the complete form of EIDG as follows:

$$\mathrm{EIDG}_i(x)_{x \in D^{\bar{p}}} = \mathbb{E}_{x' \sim U(D^p)} \left[(x_i - x_i') \times \Sigma_{k=1}^{m} \frac{F(x^k + h \cdot e_i) - F(x^k)}{h} \times \frac{1}{m} \right]. \tag{10}$$

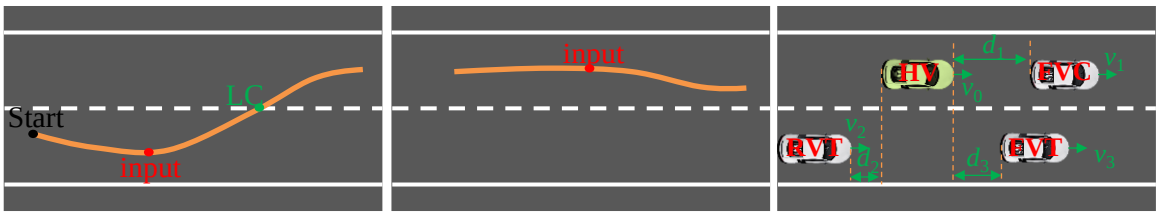
C. Axioms satisfied by EIDG

A sound feature attribution explanatory algorithm should conform to ideal axioms [23]. EIDG, an extension of the IG method in numerical differentiation, satisfies several of these ideal axioms. First, EIDG maintains *invariance*, meaning the attributions for two functionally equivalent models must be identical. If two models produce identical outputs for the same inputs, they are functionally equivalent, regardless of any variations in their internal design. In addition, EIDG maintains *completeness* by ensuring that the cumulative input attributions are equal to the difference between the model's output at the input and the baseline. This guarantees that the model output is attributed to the input features, as detailed in (5). Third, the EIDG framework adheres to the principle of *sensitivity*, meaning that if the model is independent or unrelated to the input, then the attributions of the input features should be mathematically zero. In addition, EIDG introduces a new axiom, *positivity*, which states that for a given decision, the sum of the contributions of all features should be consistently greater than or equal to zero.

EIDG complements these axioms as an appropriate explanation algorithm, providing attribution explanations for various structured machine-learning models, targeting data-driven autonomous driving.

Theorem 1: If single baseline x' is obtained from (8), then for $x \in \mathbb{X}$ and its corresponding selected decision a^*, the $\sum_{i=1}^{n} \mathrm{IG}_i(x, a^*) \geq 0$.

Proof: If single baseline x' is obtained from (8), according to the *completeness* of EIDG, then for $x \in \mathbb{X}$ and its corresponding selected decision a^*, $\sum_{i=1}^{n} \mathrm{IG}_i(x, a^*) = F_{a^*}(x) - F_{a^*}(x') \geq \max_{a \in \mathbb{A}} F_a(x) - \min_{x \in \mathbb{X}} \max_{a \in \mathbb{A}} F_a(x) \geq 0$. ■



(a) (b) (c)

Fig. 2: Data Extraction and Feature Selection for Lane-Change/Non-Lane-Change. (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.

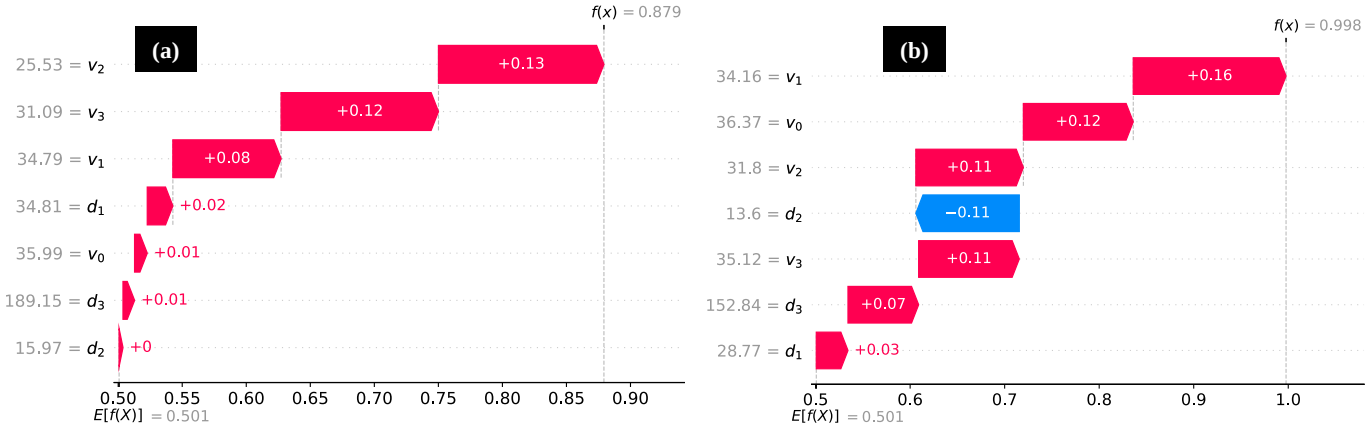


Fig. 3: Visualization of the attribution explanations for two lane-changing sample.

IV. Experiments

In this section, we deploy EIDG and validate attribution explanations on two typical autonomous driving tasks. Additionally, we conduct experimental comparisons with typical model-agnostic attribution methods. The experimental setup is as follows:

- Processor: Intel(R) Core(TM) i7-10875H CPU 2.30GHz (2.30 GHz).
- GPU: NVIDIA GeForce RTX 2060
- RAM: 40.0GB.

In the following example, the EIDG parameters are both set as $m = 100$, $h = 0.1$, and $p = 10$.

A. Feature attribution explanation for lane change decision model

We trained a lane-change (LC) prediction model on the HighD dataset [24] using a XGBoost model. While constructing this model, we followed steps and rules as [25], [26]. Specifically, for vehicle ID (58) (shown in Fig. 2 (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 as shown in Fig. 2 (b)), the 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, 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. 2 (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_{LC} (a_{LH}) represents the LC/No LC decision.

To obtain the reliable XGBoost-based LC decision model, the grid search method is used to optimize the key parameters. The final parameters are shown in Table I. Using the proportion of correct predictions among the total predictions as the evaluation metric, the model's accuracy has reached 97.8% after training.

TABLE I: PARAMETERS OF THE XGBoost FOR LATERAL MODEL

Parameter	Value	Parameter	Value
Number of estimators	60	$L1$ regularization	0
Maximum tree depth	10	$L1$ regularization	0.5

In the LC sample illustrated in Fig. 3(a), features $v_2 = 25.53$ m/s and $v_3 = 31.09$ m/s are identified as the most influential, with respective contributions of 0.13 and 0.12. The relatively lower velocity (v_2) of the rear vehicle in the target lane compared to the ego vehicle's velocity (v_0) mitigates the risk of collision when the ego vehicle changes lanes. Conversely, the velocity of the front vehicle (v_3) in the target lane closely approximates the ego vehicle's velocity (v_0), resulting in a diminished risk of collision when changing lanes.

For the LC sample depicted in Fig. 3(b), $v_1 = 34.16$m/s and $v_0 = 36.37$m/s emerge as the most influential features for the current decision, contributing 0.16 and 0.12, respectively. Notably, the lower velocity of the front vehicle (v_1) relative to the ego vehicle's velocity (v_0) adversely affects the ego vehicle's traffic efficiency, prompting the leading vehicle to change lanes to optimize driving efficiency.

B. Feature attribution explanation for vehicle following model

This section employs an Intelligent Driver Model (IDM) [27] and two various machine-learning models fitted to data generated by the IDM model to illustrate that even when the model's performance appears to align closely with actual values on specific datasets, it may still possess shortcomings. These shortcomings will be elucidated through the EIDG method, providing insights for developers and users.

1) IDM: The IDM describes the longitudinal motion state of vehicles, with its fundamental equation as follows:

$$a = \frac{dv}{dt} = a_{max} \cdot \left(1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*(v)}{s}\right)^2\right) \tag{11}$$

where v represents the vehicle's velocity, a_{max} signifies the vehicle's max acceleration, v_0 refers to the desired free-flow velocity, δ is an adjusting parameter that impacts the nonlinearity of velocity response, s represents the distance between the vehicle and leading vehicle, and $s^*(v)$ is the safety following distance function calculated based on the vehicle's velocity, expressed as:

$$s^*(v) = max\{0, s_0 + v \cdot T + \frac{v \cdot \Delta v}{2 \cdot \sqrt{a_{max} \cdot b}}\} \quad (12)$$

where, s_0 represents the vehicle's minimum stopping distance, T stands for the driver's reaction time, Δv is the relative velocity of the vehicle concerning the preceding vehicle, and b denotes the vehicle's comfortable deceleration. The variables of the IDM model can be categorized into fixed model parameters and external input variables. In our example, the fixed parameter values are shown in Table II. The input variables are $x = [s, v, \Delta v]$.

TABLE II: PARAMETERS OF THE IDM

Parameter	Value	Unit	Parameter	Value	Unit
a_{max}	4	m/s^2	T	1	s
v_0	20	m/s	s_0	10	m
δ	–	6	b	4	m/s^2

2) Data Fitting using XGBoost [28] and Deep Learning: We employed the sklearn library to construct two machine learning models, as detailed in [29]. Specifically, a fully connected 4-layer neural network is trained, with all relevant parameters detailed in Table V. The training details for the XGBoost model are provided in Table IV.

TABLE III: PARAMETERS OF THE DEEP LEARNING FOR LONGITUDINAL MODEL

Parameter	Value	Parameter	Value
Network layer	3×128×128×1	Learning rate	0.001
Optimizer	SGD	Hidden layer	ReLU

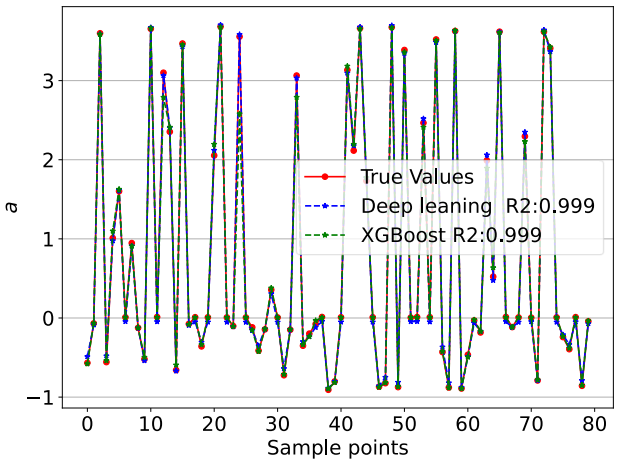


Fig. 4: Comparison of prediction between IDM (true value), XGBoost and deep learning model

TABLE IV: PARAMETERS OF THE XGBoost FOR LONGITUDINAL MODEL

Parameter	Value	Parameter	Value
Number of estimators	100	$L1$ regularization	0
Maximum tree depth	3	$L1$ regularization	0.5

3) Performance and analyze: As shown in Fig 4, it is evident that both the XGBoost and deep learning models have yielded impressive regression results with the R-squared (R2) of 0.999. However, a crucial question remains:

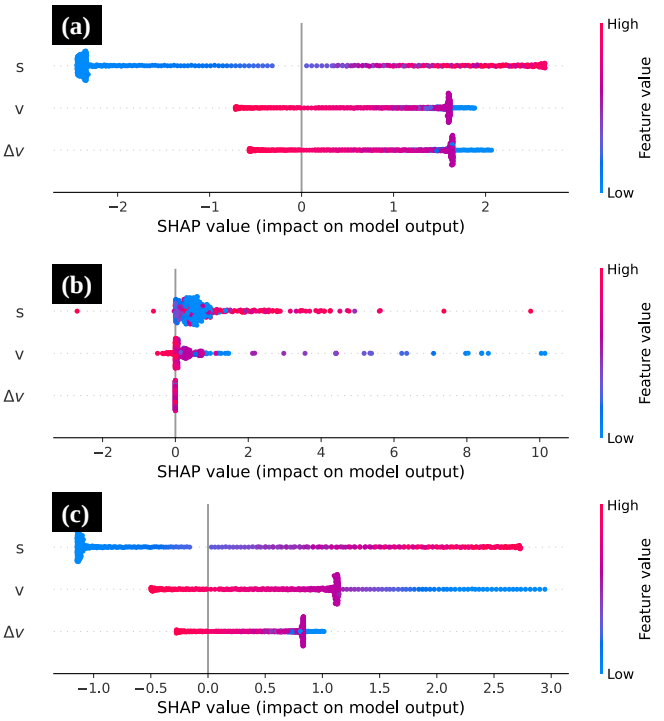


Fig. 5: **Summary of attribution values for different models.** a. For the IDM model. b. For the XGBoost model. c. For the deep learning model

Have these models truly learned the right knowledge? and which model is better?

To answer this query, we conducted a comparative experiment. Initially, we employed EIDG to compute attributions for three categories of models. Subsequently, we visually presented the distributions of attribution values corresponding to all features. As shown in Fig 5, the x-axis represents specific attribution values, and the y-axis represents the categories of features in order of importance. The dots indicate all samples, and the feature values are indicated from small to large by the corresponding blue to red color. It is evident that in the case of the IDM model illustrated in Fig. 5 (a), there is a general positive correlation between the feature values of s and their corresponding attribution values. Conversely, an approximate negative correlation exists between the feature values of v, Δv, and their respective attribution values. This observation aligns with the characteristics of the IDM model, as described in (9) and (12), where more considerable relative distances s tend to

result in increased acceleration. In contrast, higher velocities v and relative velocities Δv tend to decrease acceleration. In contrast, in the XGBoost model depicted in Fig 5 (b), the maximum attributed values for s and v greatly exceed that in the IDM model, which surpass the maximum acceleration. Moreover, the value of Δv consistently contributes zero to the acceleration, which is evidently incorrect. For the deep learning model represented in Fig. 5 (c), the trend of attribution values resembles that of the IDM model. This implies that the model has largely captured the intrinsic prediction mechanisms of the IDM model. These findings have arrived at an initial answer to the initial question:

The XGBoost model has barely grasped the IDM model's internal knowledge, while the deep learning model has largely assimilated it. Deep learning model is better than XGBoost.

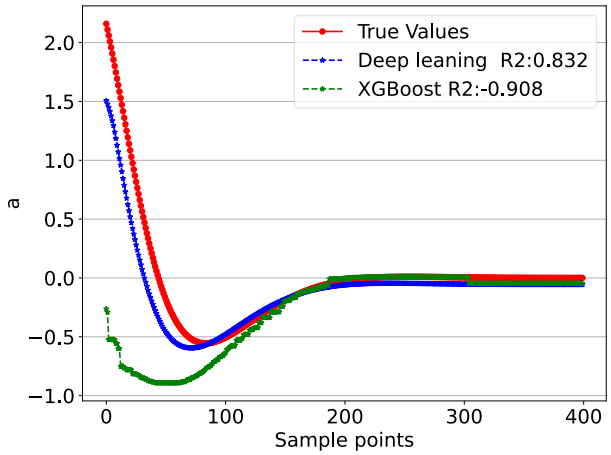


Fig. 6: Comparison of prediction between IDM (true value), XGBoost and deep learning model on D_{new}

This answer is derived from the attribution values within the existing dataset D, offering the advantage of not requiring additional datasets for testing. This is particularly beneficial for data-expensive machine learning problems. On the other hand, it is not impossible to encompass all real-world data exhaustively. To further validate the findings shown in Figs 5 and the initial answer, we tested the models on a new dataset D_{new}. The dataset was collected through a set of closed-loop car-following tests, conducted under the following conditions: the initial distance between vehicles was set at 30 m, with the preceding vehicle maintaining a constant speed of 10 m/s. The host vehicle's initial speed was set at 10 m/s. An IDM controller was utilized for the host vehicle. We tested two machine learning models on D_{new}, where the deep learning model achieved an R2 of 0.832, while the XGBoost model had an accuracy of -0.908. This indicates that the deep learning model exhibits better performance compared to XGBoost on the D_{new}. Furthermore, we have conducted an analysis of the attribution values distribution of the models on the new dataset shown in Fig 7. It is evident that the deep learning model still demonstrates a similar trend in attribution values distribution to the IDM. However, for the XGBoost model, it is apparent that relative speed has no effect on the output, confirming the preliminary findings mentioned earlier.

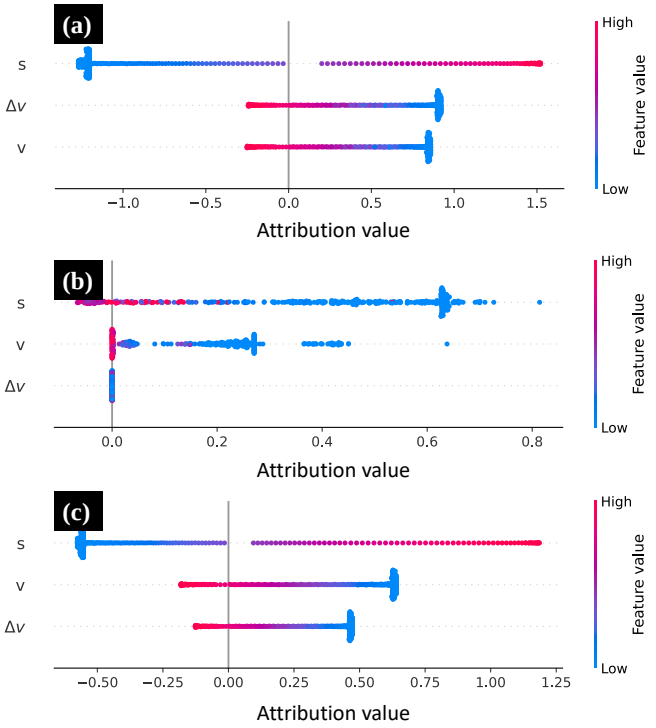


Fig. 7: **Summary of attribution values for different models on D_{new}.** a. For the IDM model, b. For the XGBoost model, c. For the deep learning model.

In summary, EIDG provides a way to explain and diagnose models beyond model accuracy metrics, assisting researchers and developers to diagnose possible defects in models for further optimization, especially in data-expensive machine learning, such as data-driven autonomous driving models.

TABLE V: 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

C. Quality of EIDG

To validate the superiority of the proposed method, we assess it from the following three aspects: 1) Scope of applicability, which measures the range of models to which it can be applied; 2) Baseline selection, evaluating the superiority of the selected base values on attribution results; and 3) Computational Efficiency, quantifying the algorithm's time consumption; 4) Numerical stability, evaluating the impact of different parameter h on the attribution results.

1) Scope of Applicability: EIDG is built upon the foundation of IG, and it is evident that our method extends its applicability to arbitrary machine learning models.

2) Baseline selection: Tasking the XGBoost model for longitudinal car-following as an example, We compared the

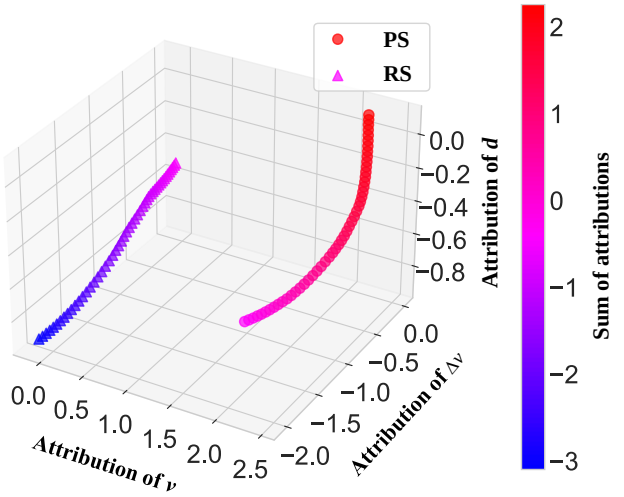


Fig. 8: The comparison between positive baselines (PS) and random baselines (RS) involves the attribution values of features. The three dimensions represent the attribution values of each feature, while the color gradient of the curves indicates the numerical magnitude of the sum of the feature attribution values, transitioning from blue to red to signify increasing values.

disparities between positive and random baselines in attribution values. As shown in Fig 8, the three dimensions respectively represent attribution values for speed, relative speed, and relative distance. In contrast, the color gradient (from blue to red) represents the sum of the attribution values, with warmer colors indicating larger values. Upon comparison, we noted that attribution values derived from the positive baselines demonstrate a notably larger magnitude. This heightened discernibility aids in identifying influential features more effectively. Moreover, when graphing the attribution values from the fixed baseline, we consistently observed them to be greater than or equal to zero, illustrating their consistent directionality.

TABLE VI: COMPARISON OF COMPUTATION TIME FOR DIFFERENT ALGORITHMS (Please refer to the Appendix A for KernelSHAP and LIME.)

Method	Convergence time sec (sample number = n)				
	$n = 10$	$n = 50$	$n = 100$	$n = 200$	$n = 300$
EIDG	0.118	0.167	0.180	0.199	0.228
SHAP [21]	0.457	2.12	4.31	8.380	12.953
LIME [19]	10.588	54.216	107.197	212.315	329.105

3) Computational efficiency: We compared our method with model-agnostic feature attribution techniques KernelSHAP and LIME. As demonstrated in Table VI, the EIDG method exhibits a significant advantage in computational

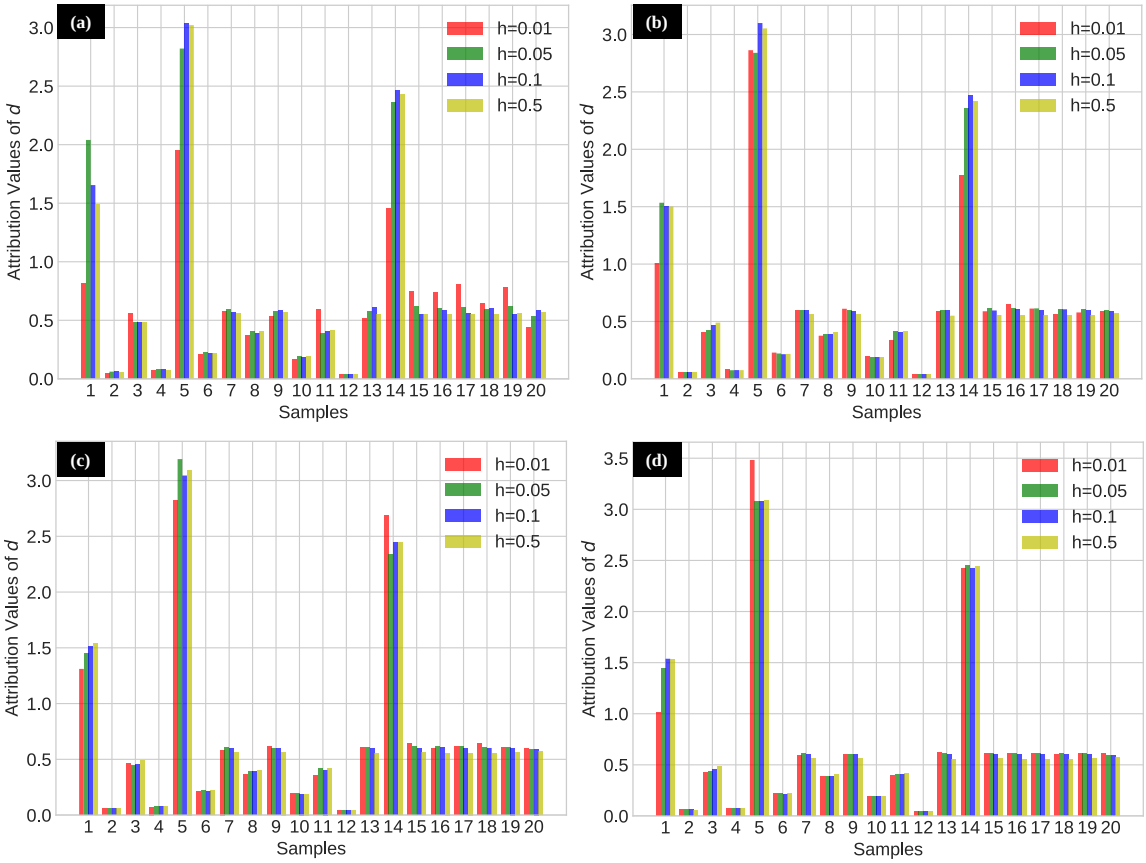


Fig. 9: **The comparison of the attribution values of relative distance d in the longitudinal model across different h**, a. For m=50, b. For m=100, c. For m=200, d. For m=400.

TABLE VII: COMPUTATION TIME OF EIDG WITH DIFFERENT PARAMETERS

Baseline number	Convergence time sec (total number of steps = m)				
	$m = 10$	$m = 50$	$m = 100$	$m = 200$	$m = 300$
p=10	0.102	0.110	0.218	0.220	0.252
p=50	0.303	0.448	0.634	0.869	1.053
p=100	0.605	0.901	1.055	1.580	2.116
p=200	1.218	1.556	2.001	2.965	3.910
p=300	1.714	2.386	3.052	4.504	5.603

efficiency with an increasing sample size.

Further investigation was conducted to explore the impact of different parameter configurations on the computational efficiency of EIDG. As illustrated in Table VII, when computing feature attribution values for 320 samples concurrently, the computation time gradually increases with the increase of p and m. However, rather than exhibiting a doubling effect, the increase in computation time is observed to be relatively modest.

4) Numerical stability: Given that EIDG employs numerical differentiation to obtain approximate gradients of the model, it is essential to analyze the impact of h values on the attribution results. Here, we collected attribution values of the relative distance feature d across different h values in 20 samples of the longitudinal model. Further analysis was conducted under various integration path lengths m. As depicted in Fig. 9, significant differences in attribution values were observed under $m = 50$ for different h values. However, as m increased, these differences gradually diminished. This insight guides us to mitigate the influence of small increment h on numerical differentiation by increasing m in practical applications.

V. Conclusion

In this study, we proposed EIDG for explaining and diagnosing machine learning models, targeting scenarios in autonomous driving. EIDG extends the traditional neural network-specific IG method to diverse structured machine learning models through numerical differentiation. It enhances the previous single baseline approach with a distributed multiple baseline scheme with a positivity constraint. We initially verified EIDG's computational efficiency using synthetic models and datasets, underscoring its capability to ensure positive attribution values. Moreover, EIDG enhances feature selection by amplifying attribution value magnitudes. Furthermore, we compared attribution values on the classic IDM model and two machine learning models fitted with data generated by IDM. The results suggest that the feature attribution values obtained through EIDG can reflect model characteristics beyond the scope of model accuracy assessment. This aids developers in gaining insights into the model's functioning. Therefore, our EIDG method steps forward in explaining and diagnosing machine learning models. We also offer the EIDG tool intending to assist users in analyzing issues across various application domains of machine learning.

Explainable AI research in the field of autonomous driving is still in its nascent stage. Interactions between features in the real world can lead to attribution biases. Therefore, in future work, integrating EIDG with causal reasoning theories can offer more accurate feature attribution explanations.

References

[1] A. K. Gizzini, Y. Medjahdi, A. J. Ghandour, and L. Clavier, "Towards explainable ai for channel estimation in wireless communications," *IEEE Transactions on Vehicular Technology*, 2023.

[2] S. Roy, H. Chergui, and C. Verikoukis, "Towards bridging the fl performance-explainability trade-off: A trustworthy 6g ran slicing use-case," *IEEE Transactions on Vehicular Technology*, 2024.

[3] P. R. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs *et al.*, "Outracing champion gran turismo drivers with deep reinforcement learning," *Nature*, vol. 602, no. 7896, pp. 223–228, 2022.

[4] S. Feng, H. Sun, X. Yan, H. Zhu, Z. Zou, S. Shen, and H. X. Liu, "Dense reinforcement learning for safety validation of autonomous vehicles," *Nature*, vol. 615, no. 7953, pp. 620–627, 2023.

[5] Y. H. Khalil and H. T. Mouftah, "Exploiting multi-modal fusion for urban autonomous driving using latent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 2921–2935, 2022.

[6] X. He and C. Lv, "Towards safe autonomous driving: Decision making with observation-robust reinforcement learning," *Automotive Innovation*, vol. 6, no. 4, pp. 509–520, 2023.

[7] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[8] H. Chen, I. C. Covert, S. M. Lundberg, and S.-I. Lee, "Algorithms to estimate shapley value feature attributions," *Nature Machine Intelligence*, pp. 1–12, 2023.

[9] S. M. Lundberg, B. Nair, M. S. Vavilala, M. Horibe, M. J. Eisses, T. Adams, D. E. Liston, D. K.-W. Low, S.-F. Newman, J. Kim *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nature biomedical engineering*, vol. 2, no. 10, pp. 749–760, 2018.

[10] G. Erion, J. D. Janizek, P. Sturmfels, S. M. Lundberg, and S.-I. Lee, "Improving performance of deep learning models with axiomatic attribution priors and expected gradients," *Nature machine intelligence*, vol. 3, no. 7, pp. 620–631, 2021.

[11] A. Selbst and J. Powles, "meaningful information and the right to explanation," in *conference on fairness, accountability and transparency*. PMLR, 2018, pp. 48–48.

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

[13] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[14] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of deep vision-based autonomous driving systems: Review and challenges," *International Journal of Computer Vision*, vol. 130, no. 10, pp. 2425–2452, 2022.

[15] Q. Zhang, X. J. Yang, and L. P. Robert, "Expectations and trust in automated vehicles," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–9.

[16] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[17] R. Sayres, A. Taly, E. Rahimy, K. Blumer, D. Coz, N. Hammel, J. Krause, A. Narayanaswamy, Z. Rastegar, D. Wu *et al.*, "Using a deep learning algorithm and integrated gradients explanation to assist grading for diabetic retinopathy," *Ophthalmology*, vol. 126, no. 4, pp. 552–564, 2019.

[18] J. R. Zech, M. A. Badgeley, M. Liu, A. B. Costa, J. J. Titano, and E. K. Oermann, "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study," *PLoS medicine*, vol. 15, no. 11, p. e1002683, 2018.

[19] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[20] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *International conference on machine learning*. PMLR, 2017, pp. 3145–3153.
[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.
[22] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.
[23] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.
[24] 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 *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 2118–2125.
[25] 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 Transactions on Intelligent Vehicles*, 2023.
[26] M. Li, H. Sun, Y. Huang, and H. Chen, "Svce: Shapley value guided counterfactual explanation for machine learning-based autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2024.
[27] 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 Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1422–1428, 2016.
[28] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou *et al.*, "Xgboost: extreme gradient boosting," *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4, 2015.
[29] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

VI. Biography Section

Meng Li received the B.S. degree in 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 with the School of Automotive Studies, Tongji University, China, in 2023. He is currently pursuing the M.S. degree with the School of Automotive Studies, Tongji University, China. His research interests include machine learning, deep learning, explainable AI and autonomous vehicle.

Zhihao Cui received the B.S. degree in the School of Vehicle Engineering, Chongqing University of Technology, China, in 2021. He is currently pursuing the M.S. degree with the School of Automotive Studies, Tongji University, China. His research interests include machine learning, deep learning, explainable AI and autonomous vehicle.

Yanjun Huang is a Professor at School of Automotive studies, Tongji University. He received his PhD Degree in 2016 from the Department of MME at University of Waterloo. His research interest is mainly on improving vehicle performance in terms of safety, energy-saving, and intelligence by using advanced control and learning methods. He has published several books, over 60 papers in journals and conference; He is the recipient of IEEE VTS 2019 Best Land Transportation Paper Award, the 2018 Best paper of Automotive Innovation, etc. He is serving as AE or EBM of IET Intelligent Transport System, SAE Int. J. of Commercial vehicles, Int. J. of Autonomous Vehicle system, etc.

Hong Chen (M02-SM12) 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 the Institut fuer Systemdynamik und Regelungstechnik, University of Stuttgart. Since 1999, she has been a professor at Jilin University and hereafter a Tang Aoqing professor. Recently, she joined Tongji University as a distinguished professor. Her current research interests include model predictive control, nonlinear control, artificial intelligence and applications in mechatronic systems e.g. automotive systems.

Appendix

A. Introduction of LIME and KernelSHAP

1) LIME: Given a black-box model $f : \mathbb{R}^n \to \mathbb{R}$. For a given input x, our goal is to find a locally interpretable model $g : \mathbb{R}^n \to \mathbb{R}$ to approximate the behavior of f at x. The algorithm flow is as follows:

1 Sample N data points $\left\{x^j\right\}_{j=1}^K$ centered at x from the input space;
2 Construct a linear approximation model (which can actually be any other interpretable model, the linear model is used here as an example): $g = \phi_0 + \phi_i x_i + \cdots \phi_n x_n$, where the coefficients $\phi = \begin{bmatrix}\phi_1 & \cdots & \phi_i & \cdots & \phi_n\end{bmatrix}$ i.e., the feature-attributed values of the solution to be solved;
3 Optimization object: Use interpretable model g to fit a local approximation model to the extracted features, aiming to closely match the predictions of the original model f on nearby data points. This can be formulated as:

$$\underset{g\in\mathcal{G}}{\text{minimize}} \sum_{j=1}^{N} \left[\left(f(x^j) - g(x^j)\right]^2 \omega\left(x^j\right) + \Omega(g) \right. \quad (13)$$

where $\mathcal{G}$ is the hypothesis space for the explanation model, x^j is the weight function, and $\Omega(g)$ is a penalty term for model complexity, normally $\Omega(g)$ is set to 0.
4 Explanation Generation: Interpret the important features of the input x based on the coefficients of the model g.

2) KernelSHAP: KernelSHAP is a method for approximating Shapley values, primarily because exact computation of Shapley values requires iterating through all feature combinations, resulting in exponential time complexity as the number of features increases. The details of KernelSHAP are as follows:

1 Define the linear regression model$g$: $g(z) = \phi_0 + \sum_{i=1}^{n} \phi_i z_i$, where z denotes the binary vector of feature subsets (1 means the feature is present and 0 means the feature is absent).

2 Further, KernelSHAP uses a kernel function to define the weights of the different feature subsets, i.e:

$$w(z) = \frac{n-1}{\binom{n}{|z|}|z|(n-|z|)} \tag{14}$$

where $|z|$ is the number of elements in the vector z with value 1.

3 Finally, the ϕ are estimated by solving a weighted least squares problem:

$$\min_{\phi} \sum_{z \subseteq \{0,1\}^n} [f(h(z)) - g(z)]^2 w(z) \tag{15}$$

where $h(z)$ is the mapping function that converts the binary feature subset vector z to the actual subset $\mathcal{S}$, i.e., $\mathcal{S} = h(z)$.