Visual Evaluation for Autonomous Driving

Yijie Hou, Chengshun Wang, Junhong Wang, Xiangyang Xue, Xiaolong (Luke) Zhang,
Jun Zhu, Dongliang Wang, Siming Chen

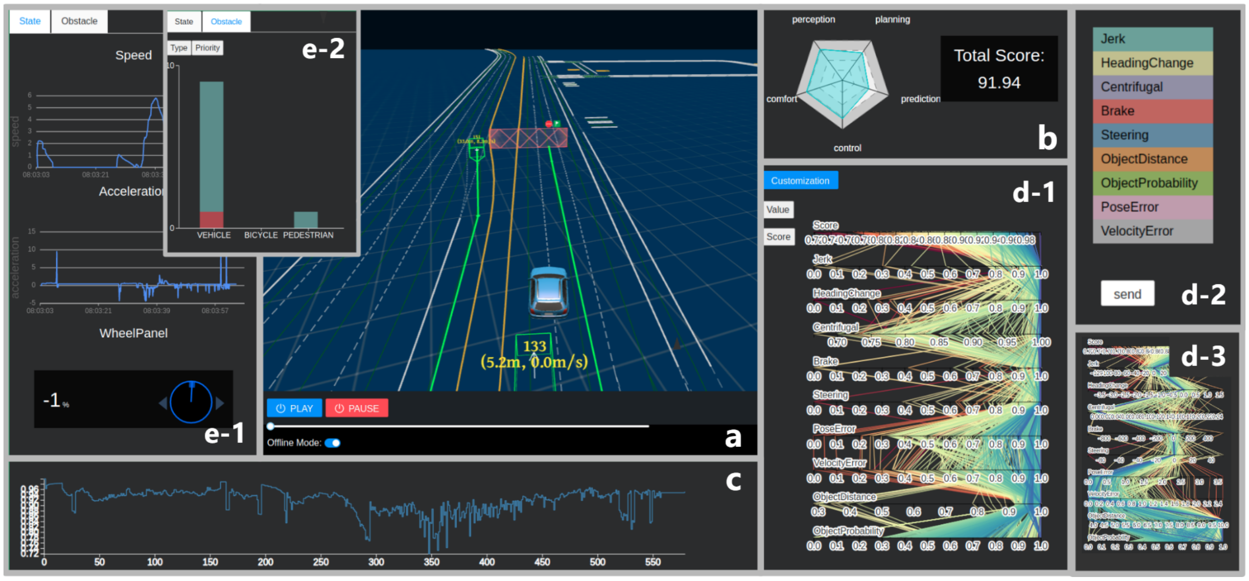


Fig. 1: System User Interface: (a) Spatial-temporal view to display a simulation scenario of autonomous driving; (b) Radar view to present the evaluation scores of five modules of autonomous driving and an overall evaluation score at a given time; (c) Timeline view to show the scores over time; (d) Parallel coordinates view to display both the score (d-1) and factor value (d-3) of each factor from five modules, with customizable ranking settings for different factor priorities (d-2); (e) Visualization of the autonomous driving states on speed, acceleration, wheel turning angle of autonomous driving vehicles (e-1), as well as the type and priority distributions of obstacles in the scenario (e-2).

Abstract— Autonomous driving technologies often use state-of-the-art artificial intelligence algorithms to understand the relationship between the vehicle and the external environment, to predict the changes of the environment, and then to plan and control the behaviors of the vehicle accordingly. The complexity of such technologies makes it challenging to evaluate the performance of autonomous driving systems and to find ways to improve them. The current approaches to evaluating such autonomous driving systems largely use a single score to indicate the overall performance of a system, but domain experts have difficulties in understanding how individual components or algorithms in an autonomous driving system may contribute to the score. To address this problem, we collaborate with domain experts on autonomous driving algorithms, and propose a visual evaluation method for autonomous driving. Our method considers the data generated in all components during the whole process of autonomous driving, including perception results, planning routes, prediction of obstacles, various controlling parameters, and evaluation of comfort. We develop a visual analytics workflow to integrate an evaluation mathematical model with adjustable parameters, support the evaluation of the system from the level of the overall performance to the level of detailed measures of individual components, and to show both evaluation scores and their contributing factors. Our implemented visual analytics system provides an overview evaluation score at the beginning and shows the animation of the dynamic change of the scores at each period. Experts can interactively explore the specific component at different time periods and identify related factors. With our method, domain experts not only learn about the performance of an autonomous driving system, but also identify and access the problematic parts of each component. Our visual evaluation system can be applied to the autonomous driving simulation system and used for various evaluation cases. The results of using our system in some simulation cases and the feedback from involved domain experts confirm the usefulness and efficiency of our method in helping people gain in-depth insight into autonomous driving systems.

1 Introduction

Autonomous driving technologies have advanced rapidly in recent years. More and more car manufacturers consider the installation of autonomous driving systems in their cars, in particular those electronic models. By letting people stay away from the steering wheel, autonomous driving can reduce the stress associated with driving. However, delegating decision-making in driving to algorithm-driven automatic systems also raises many concerns, such as driving safety and

- *Y. Hou, C. Wang, J. Wang, X. Xue and S. Chen are with School of Data Science, Fudan University. Y. Hou and C. Wang contribute equally. S. Chen is the corresponding author. E-mail: {yijiehou, chengshunwang, junhongwang, xyxue, simingchen}@fudan.edu.cn.*
- *X. Zhang is with Pennsylvania State University. E-mail: lzhang@ist.psu.edu.*
- *J. Zhu and D. Wang are with China FAW (Nanjing) Technology Development Co., Ltd. E-mail:{zhujun18, wangdongliang}@faw.com.cn.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

algorithm reliability [1].

Autonomous driving model developers and researchers have investigated methods to evaluate autonomous driving technologies to identify design deficiencies and then fix them. Recently, various evaluation criteria for autonomous driving have been proposed. For example, the Society of Automotive Engineers (SAE) released a classification to categorize autonomous driving technology into six levels, from L0 to L5 [2]. China also published a white paper with its own classification, from T1 to T5 [3]. These classifications grade autonomous driving vehicles based on their performances in multiple driving tasks in various scenarios. Although these qualitative evaluation criteria provide a clear understanding of what an autonomous driving system can achieve, they cannot tell what factors may contribute to success or failure in a test. For example, when an evaluation method can downgrade a system due to its failure in a test of turning at an intersection (e.g., failing to yield to a vehicle or a pedestrian), recognizing the causes of the failure requires more advanced techniques to understand the interior of the system.

Some research has explored quantitative evaluation methods to reveal more details of autonomous driving systems. However, some of these methods evaluate autonomous driving from the perspectives of task complexity and environmental complexity without considering objective metrics, such as starting acceleration and following distance [4], while some rely solely on driving experience or conventional autonomous driving scenarios for evaluation model selection without the integration of official evaluation guidance or important criteria provided by autonomous driving module developers [5–7]. Many large technology companies have developed autonomous driving visualization tools, such as Apollo Dreamview [8], etc. However, they merely visualize data from autonomous driving modules, not evaluate them.

We propose a visual analytics approach for the evaluation of autonomous driving systems. Our goal is to let module developers understand the performance of the autonomous driving system. We aim to provide module developers with tools to investigate and explain various evaluation scores of an autonomous driving system through an intuitive and interactive interface. We implement a visual analytics system to support the user-driven computation, exploration, and evaluation of the performance score for each factor in each module of the whole autonomous driving process. Through various visualization designs, module developers can examine the overall performance of the system, the performance of each involved module, and the impacts of individual data factors. Our research contribution is as follows:

- We develop **a visual evaluation workflow** from overview to details that helps autonomous driving module developers explore the performance of each module and related contributing factors.
- We build **a visual evaluation system** to support visual and interactive evaluation of the whole process of autonomous driving.
- We develop **application scenarios for autonomous driving evaluation** with cases evaluated by the domain experts.

The paper is structured as follows. Sect. 2 reviews related work. Then, we briefly describe the workflow of autonomous driving and derive the evaluation requirement in Sect. 3. Our visual evaluation method is presented in Sect. 4. To demonstrate the capability of our approach, we introduce the applications of our method in the simulation driving datasets in Sect. 5 and our evaluation work with domain experts in Sect. 6. Finally, we discuss the limitations and future work in Sect. 7.

2 Related Work

2.1 Autonomous Driving and Simulation

The success of Boss, the champion vehicle the DARPA Urban Challenge [9], has ignited autonomous driving and also popularized the software structure of its autonomous driving system, which include such major subsystems as perception, motion planning, and decision-making. Nowadays, various types of autonomous driving systems have been developed. A recent comprehensive summary on the development of autonomous driving by Yurtsever et al. [10] discusses the challenges in autonomous driving and presents available databases and tools during the development of the autonomous driving system. In addition to technical challenges, there are also many social concerns in autonomous driving that have been widely discussed. Litman [11] explores the benefits and costs of autonomous driving as well as its development in transport planning, transport safety, energy-saving, and emission reduction, and other social issues in the future.

An autonomous driving system must go through rigorous tests before being put on the road. Considering the potential high costs of testing on public roads, research on simulation environments before road tests has become an important topic. Many corporations or organizations provide open-source autonomous driving frameworks such as Autoware [12] and Apollo [13] and continuously develop and improve autonomous driving simulation tools. Various simulation tools have been developed. CARLA [14, 15], an open-source autonomous driving simulation tool, supports training, prototype design, perception, control, and other autonomous driving model verification, and provides available signals for driving strategy training and a variety of environmental conditions specified. LGSVL [16], a high fidelity simulation tool for the autonomous driving, supports customizable simulations by allowing the creation of new controllable objects and the replacement of simulation modules. In addition, corporations are also active in developing their own testing methods. For example, Baidu has proposed its simulation test platform, Dreamland, within its autonomous driving platform Apollo [17]. Dreamland provides various simulation scenarios of autonomous driving performances (e.g., collision detection).

These tools and platforms largely focus on data generation with a visible scene and the presentation of evaluation results, and do not support user-driven in-depth evaluation of an autonomous driving systems. However, they can be used as the foundations for interactive visual analytics of the performances of the system and individual modules. The data generated by LGSVL can be used to gain the insight into the behaviors of autonomous driving system. Apollo Dreamview [8], as a visualization platform for autonomous driving data, can be used to develop our visual analytics evaluation method of the performance scores and contributing factors. To our best knowledge, our method is the first of its kind to use a visual evaluation workflow to analyze the key components of a whole autonomous driving process.

2.2 Evaluation methods for Autonomous Driving

Currently, the evaluation of autonomous driving is mainly divided into qualitative evaluation and quantitative evaluation. Qualitative evaluation methods can directly show the quality of autonomous driving technology but face challenges in explaining the rationale and evaluation criteria of autonomous driving decisions. For example, while we can easily tell that an autonomous driving system fails to avoid a pedestrian in a test, finding the true cause of the failure (e.g., a high speed or an insufficient deceleration distance) is a non-trivial task.

Quantitative evaluation is important to the evaluation and analysis of autonomous driving but currently lacks unified standards. Researchers often use their own methods [18]. Some have chosen objective indicators based on their experience and expert opinions. Even so, the parameters they choose are quite different. Meng et al. [5] divide autonomous driving data into three parts, intersection behavior, objective avoiding behavior, and car-following behavior. They evaluate the autonomous system based on parameters generated from these three parts. Dong et al. [6] select such parameters as driving time, detection of signs and lines, velocity variance in evaluation. Evaluation models include Grey Relational Analysis, AHP, TOPSIS, or just by subjective experience [5–7]. Different choices on parameters lead to different mathematical modeling methods (e.g., information-theory-based vs. entropy-based), so the models of these projects cannot be directly compared or transferred. Most of evaluation methods discussed above are mathematical and lack interactive or visualization support.

Visual analytics approaches are argued to be suitable for complex situation awareness [19, 20] and for explanation of machine learning models [21]. However, research on visual analytics in autonomous driving is rare. One exception is VATLD [22], but its focus is on a specific component of an autonomous driving system, traffic light detection, rather than the whole process. More research is needed in this area.

2.3 High-dimensional and Spatial-temporal Visualization

In autonomous driving evaluation, data of interest is usually high-dimensional spatial-temporal information. Thus, we review relevant literature on the visualization of such data, which has been an important topic in visualization research [23–25].

High-dimensional data visualization mainly include dimension reduction algorithm, subspace partition, interactive visualization customization, parallel coordinates, radar view, etc. [26, 27]. Dimension reduction (DR) methods are mainly divided into linear mapping and nonlinear mapping. The general DR methods are PCA, LDA [28], and the kernel-based methods KPCA, KFDA [29, 30], and flow learning such as Isomap, LE, and LLE [31, 32]. However, because the DR method lacks explainability, we mainly adopt the visualization methods like parallel coordinates and radar view to visualize the high-dimensional data in autonomous driving evaluation.

Autonomous driving is a good application scenario for spatial-temporal visual analytics. Regarding the visualization of spatial-temporal data, Andrienko et al. surveyed available software tools [33], and also proposed an approach to reveal patterns and trends of mass mobility through spatial and temporal abstraction of movement data [34]. Ferreira et al. [35] supported spatio-temporal queries for taxi data and enabled the examination of mobility across the city. Chen et al. [36] proposed an visual analytical workflow using context information for real-world vehicle trajectory analysis to identify dangerous driving behaviors. In our work, the high-dimensional and spatial-temporal data comes from the various components of an autonomous driving system, and the characteristics and patterns of such system data differs from those of data generated from human behaviors seen in those systems mentioned above.

3 Overview

We collaborate with a group of experts from a car company who focus on the development of autonomous driving systems. We have in-depth collaboration in the requirement specification, visualization justification and expert evaluation. Our general goal is to support the visual evaluation of autonomous driving systems. In this section, we first summarize the components in an autonomous driving system and derive the suitable data for evaluation. Then, we describe the design requirements for a visualization-based evaluation system.

3.1 Autonomous Driving Framework

Generally speaking, an autonomous driving system include three major components: perception, decision-making, and control. The *Perception* system concerns the use of various sensors that collects basic data on a vehicle and its surrounding environment. Currently, involved sensors often include GPS, IMU, ultrasonic radar, millimeter-wave radar, and cameras. Information collected from these sensors is diverse and can include essential information about the vehicle (e.g., its position, orientation, and driving mode), signal information (e.g., traffic lights), surrounding obstacles (e.g., obstacle category, position), etc.

The results of the perception component are the inputs to the *decision-making* component. Combined with non-environmental factors such as traffic rules and driver experience, the decision-making component predicts the changes of the driving environment and completes the judgment on the behaviors of the traffic participants and the computation of their potential travel trajectories.

In the autonomous driving mode, the results of the decision-making component are used to *control* the actions of the vehicle. First, a travel path is calculated based on the results, and then a corresponding driving action plan that includes a series of instructions is produced. To execute the action plan, the autonomous driving system transmits the instructions to the vehicle body to control various systems of the vehicle, including braking, steering, engine, and signals. During this process, the autonomous driving system continuously monitors the status of the vehicle and the surrounding environment, and if necessary, adjusts the action plan.

Based on the above autonomous driving process and design requirements from module developers we collaborate with, we classify autonomous driving data into five modules: perception, planning, prediction, control, and comfort. Our design to support the evaluation of autonomous driving systems targets the evaluation of these modules.

3.2 Data for Evaluation

Autonomous driving data can be obtained from the real car tests or simulation tests. Due to the limitations of the real testing environment, we use the autonomous driving data records obtained from the LGSVL simulator in this work. Based on the official classification [3] and the inputs from our collaborators, we divide our data into five modules: **perception, planning, prediction, control, and comfort**. In each module, there are several measurement *factors*, *i.e.*, the attributes captured by the sensors, that are related to its performance. The measurement factors of the **perception module** are from the perception component, and include signal detection accuracy, obstacle detection accuracy, as well as the accuracy of distance between vehicle and obstacles. With the perceived information, the autonomous driving system needs to predict the behaviors of obstacles and traffic participants. The **prediction module** from the decision making component can be measured with the accuracy of the predicted trajectory of obstacles. For the **planning module** with the understanding of the environment and obstacles, which is also from the decision making component, autonomous driving model developers are interested in such measures as the differences between predicted vehicle speed and the actual speed at a given time, the differences between predicted position and actual position, etc. The **control module** from the control component includes those factors related to the control of throttle, brake, steering wheel, etc. The **comfort module**, which concerns the feelings of the driver, uses measures like the acceleration rate of the vehicle, turning angle, etc. Table 1 summarizes these modules, involved factors in each module, their descriptions, and their evaluation criteria.

3.3 Design Requirements

Our design requirements for the visual evaluation system are developed based on the real-world application needs from our industrial collaborators. Our visual evaluation system is designed to provide the module developers of autonomous driving systems with a perceptible and interactive evaluation system so that they can better understand and optimize the overall performance of an autonomous driving system. More specifically, the requirements can be summarized as the following:

- **R1**: Module developers should be able to use the system to assess the overall performance of an autonomous driving system with a score as general feedback.
- **R2**: The system should allow module developers to evaluate the performance of a system in different time periods of the whole driving process and to identify the time periods with bad performance.
- **R3**: The system should provide methods for the evaluation of the five modules (perception, prediction, planning, control and comfort) so that module developers can tell the performances of individual modules.
- **R4**: The system should allow module developers to observe the performances of individual module at any time period and to identify those factors that contribute to the observed performances.
- **R5**: The system should allow module developers to customize the ranking of importance of each module based on their different goals and contexts in analysis.

4 Visual Evaluation Approach

This section describes our method. Observing autonomous driving data, we can determine the factors for autonomous driving evaluation based on the previous evaluation criteria and driving experience. With the help of AHP [37] and TOPSIS [38], two mathematical evaluation models, we evaluate and analyze each factor, and the evaluation result is obtained by weighted calculation. Finally, we introduce the visual interface design.

Table 1: Evaluation factor with its description and criteria

Module	Factor	Description	Evaluation Criteria
Comfort	Jerk	Acceleration change rate	The smaller the absolute value, the better; less than 1 is the positive ideal value
	Heading Change	Orientation change rate	The smaller the absolute value, the better; less than 1 is the positive ideal value
	Centrifugal	Centrifugal fore, calculated as $speed^2 \times kappa$	The smaller the absolute value, the better; less than 1 is the positive ideal value
Control	Brake	Brake strength change rate	The smaller the absolute value, the better; less than 1 is the positive ideal value
	Steering	Steering strength change rate	The smaller the absolute value, the better; less than 1 is the positive ideal value
Planning	Pose Error	The difference between positions information planned and actual positions information at the corresponding time point	The smaller the absolute value, the better; the ideal value is 0
	Velocity Error	The difference between velocity information planned and actual velocity information at the corresponding time point	The smaller the absolute value, the better; the ideal value is 0
Perception	Object Distance	The shortest distance between autonomous driving vehicle and obstacles	Combined with vehicle speed, the larger the value taken, the better. If speed is less than 10m/s, the optimal value is greater than 20; if speed is greater than 10m/s. the optimal value is greater than speed-10
Prediction	Probability	The accuracy between the predicted and the actual trajectory of obstacles	The bigger the value, the better; the optimal value is 1

4.1 Mathematical Modeling

4.1.1 Analytic Hierarchy Process (AHP)

The basic idea of AHP is to stratify the evaluation decision problem. According to the design requirements and description, the problem is decomposed into different component factors. Based on the comparison of these factors and their affiliations, the factors are cohesively combined at different levels to form a multi-level analysis structure model. Finally, the objectives in the problem are compared and ranked according to their performance in the model.

AHP uses the consistent matrix method to construct the judgment matrix. Let r_{ij} denote the importance of factor i relative to j, which ranges from 1 (factor i is as important as j) to 9 (factor i is much more important than j); then the importance of factor j relative to i is $r_{ji} = \frac{1}{r_{ij}}$. Fill the scale value r_{ij} into the matrix R to get the judgment matrix. When the judgment matrix R satisfies the consistency test, the eigenvector corresponding to its maximum eigenvalue λ_{max} is $\omega = (\omega_1, \omega_2, \cdots, \omega_n)^T$. According to the properties of consistency matrix, we have $r_{ij} = \frac{w_i}{w_j}$. Based on the construction of the judgment matrix, ω_i and ω_j can be taken as the absolute importance of factor i and factor j. Finally, the weights of the factors are obtained by normalizing ω.

4.1.2 Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS)

The basic principle of the TOPSIS method is to rank the evaluation objects from the positive ideal solution to the negative ideal solution by calculating the distances between them. If the evaluation object is close to the positive ideal solution and far away from the negative ideal solution, it is optimal. The factor values of the positive ideal solution and the negative ideal solution represent the positive ideal value and the negative ideal value of each evaluation factor, respectively.

In our case, we divided the autonomous driving process into five modules and selected relevant evaluation factors in each module. We combine AHP and TOPSIS in calculating the overall evaluation score for autonomous driving, because there are no officially defined evaluation criteria for autonomous driving. Therefore, with the help of TOPSIS, for a driving record over a period of time and a given evaluation factor, a time series of evaluation results for the factor can be obtained. In this way, we can obtain the time series of evaluation results for all factors involved in the evaluation of autonomous driving over a period of time. Because there are no standard rules for ranking the importance of the factors of autonomous driving, AHP is used to determine the weight of each evaluation factor. Initially the weights are determined by module developers to equalize every evaluation module weight. With our visual evaluation system, they can determine the importance of each factor for different needs by themselves and the system can calculate the parameters that meet their needs with the help of AHP.

Assuming that there are n factors for evaluation, the matrix $A^{m \times n} = \{a_{ij}\}$ represents the autonomous driving data for n factors at m time points, and the evaluation criterion is $r = (r_1, r_2, \cdots, r_n)$. Based on TOPSIS, the positive ideal solution A^+ and the negative ideal solution A^- can be determined. The criteria for determining the positive and the negative ideal solutions vary by factors, as shown in Table1. For each object, its distances to positive ideal solution D^+ and negative ideal solution D^- are calculated separately. The relative closeness can be calculated by $B = \frac{D^-}{D^+ + D^-}$. The higher the relative closeness is, the better the evaluation result will be. The overall evaluation result matrix $S = (B_1, B_2, \cdots, B_n)$ can be obtained. We can then calculate the final weighted evaluation matrix $Z = Sdiag(\omega)$ with normalized evaluation weights ω. In our case, we set the importance of each module equal at the beginning. Module developers can determine the relative importance of factors and rank them to generate new criteria r^* and judgment matrix R^*. Repeating these steps leads to a new evaluation result Z^*, which satisfies the user's evaluation needs.

Due to our computing design, there must be at least one negative ideal solution for each factor, *i.e.*, each evaluation factor has at least one object that takes the value of 0 even though this evaluation factor may have an excellent overall evaluation result. This design amplifies the evaluation difference between values of the factor, which may lead to misjudgment in the evaluation. Therefore, we make adjustments in TOPSIS with the assumption that the difference between the positive ideal solution and the negative ideal solution of the evaluation factor is minimal. In such case, we can add a regulation factor γ so that the shortest distance between each object of the factor and the negative ideal solution is γ. This will produce a better evaluation result. We visualize the evaluation results in the visual interface.

4.2 Visual Interface

The visual interface, as shown in Fig. 1, mainly includes the components for score and state visualization. The score visualization component contains four parts to show the total score (R1, Fig. 1-b), overall scores over time (R2, Fig. 1-c), scores for five modules at each time (R3, Fig. 1-b. Left) and the score of factors for each module in an autonomous driving process (R4, Fig. 1-d.1). The state visualization component mainly contains the spatial-temporal simulation scene of the whole process (Fig. 1-a) to show the overview of the autonomous driving scene, which includes a map, lane lines, traffic signals, various obstacles, predicted paths of obstacles, and planned paths of autonomous vehicles. The state visualization also presents the states of

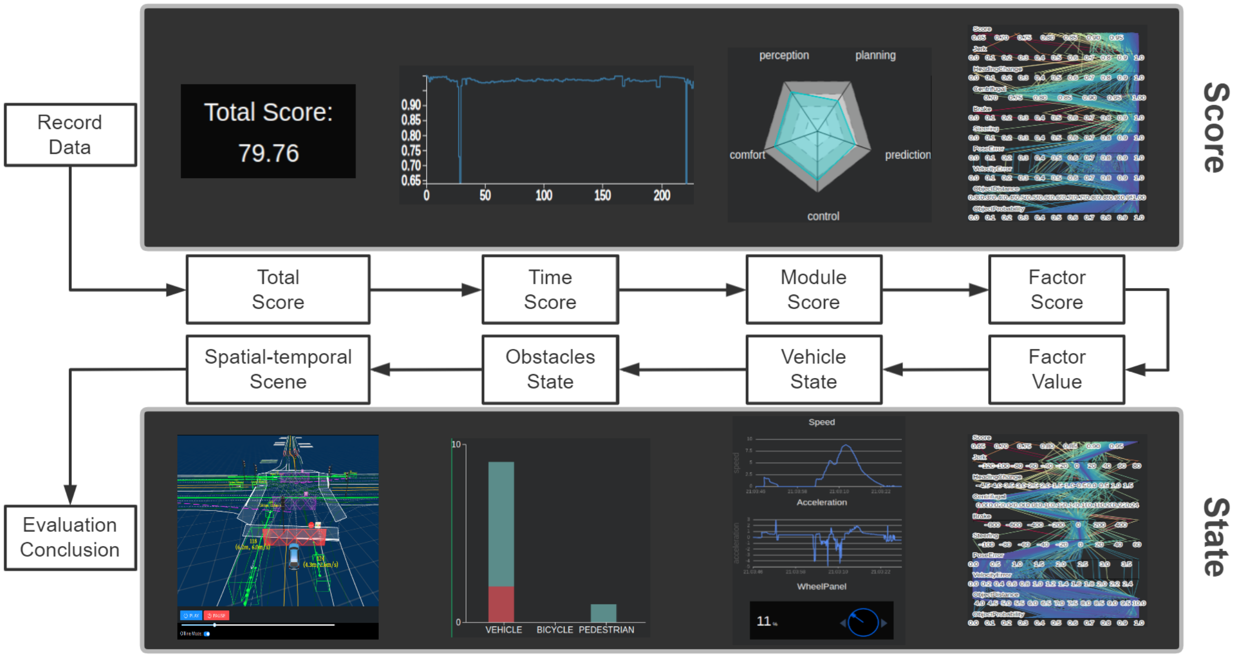


Fig. 2: The visual evaluation workflow. Our visualization system is mainly divided into into two parts, the score visualization part and the state visualization part. From overview to detail, autonomous driving module developers as users can investigate the total score, score distribution over time, module score at a specific time period and scores of the factor from the module. To understand why the score is good or not, users can interactively explore the autonomous driving state including the factor values and spatial-temporal scene.

the autonomous driving vehicle (Fig. 1-e.1) and obstacles during the process (Fig. 1-e.2), and the values of each factor involved (Fig. 1-d.3). Module developers can explore the scores from overview to details and then identify the state of each module and the spatial-temporal scenario to better understand the performance (Fig. 2).

4.2.1 Score Visualization

With the input of the autonomous driving data record, we first calculate the scores for each factor based on our model described in Sect. 4.1. The total score, module score and factor score range from 0 to 1. The score of each module is averaged by the scores of all the factors from this module at each time period. The score of each time period is averaged by the scores of all the modules at this time period. The total score is calculated by the average of all the scores in every time step. Module developers first read the total score in Fig. 1-b (R1) and examine the timeline visualization (R2, Fig. 1-c) to check the trend of the scores along the time. They can identify the outlier moments with the low scores and check details in the radar view (Fig. 1-b). We choose the radar chart because we have five modules which is a suitable number of axes for choosing the radar chart. Users can easily identify the good and bad performance scores in different modules.

From the radar view, the score distributions of five modules, including perception, planning, prediction, control, and comfort are visualized (R3). We can compare and analyze each module to observe the performance of different autonomous driving evaluation modules. The radar view shows the overall performance of the five modules at the initial moment and the module scores at specific moments. It can be dynamically updated along with the timeline. We can find the time period with poor or unbalanced evaluation scores of each module.

Each module contains several factors for evaluation. For example, the comfort module has Jerk, HeadingChange, and centrifugation (large centrifugal means that the car goes through a small radius curve at high speed and in this situation, the passengers are prone to dizziness and nausea.) of the vehicle. The specific evaluation modules and factors are detailed in Table 1.

There are four design alternatives for visualizing the scores and factor values of the five modules, including scatter plot matrix, radar chart, projection method, and parallel coordinates. Scatterplot matrix is not suitable because it is not a multivariate solution but a multiple bivariate solution. Radar chart has already been used in the module visualization (Fig. 1-b) and may be cluttered with larger amount of factors while the projection methods like PCA, t-SNE lack interpretability. Parallel coordinates provide a good display of high-dimensional data and help users perceive both scores and values of all factors. The axes of the parallel coordinates are the corresponding factors of the five modules, arranged based on the order of these modules. This can help to alleviate the order issue in parallel coordinates and to better reflect data patterns. Moreover, the space availability in the user interface can accommodate a view of parallel coordinates, and our collaborators agree on it.

Therefore, a view of parallel coordinates (Fig. 1-d) is used to show the situation of each factor involved in evaluation (R4). As shown in the figure, the top axis of parallel coordinates is the score axis, which shows the evaluation result of each time point. The polylines in parallel coordinates are color encoded by the total score at a specific moment. Under the score axis are factor axes, which indicate the evaluation result of each factor at each time point (Fig. 1-d.1). With the help of parallel coordinates, users can easily see the connection of between overall scores and the values of individual factor, and observe the influence of the evaluation factors on the modules.

4.2.2 State Visualization

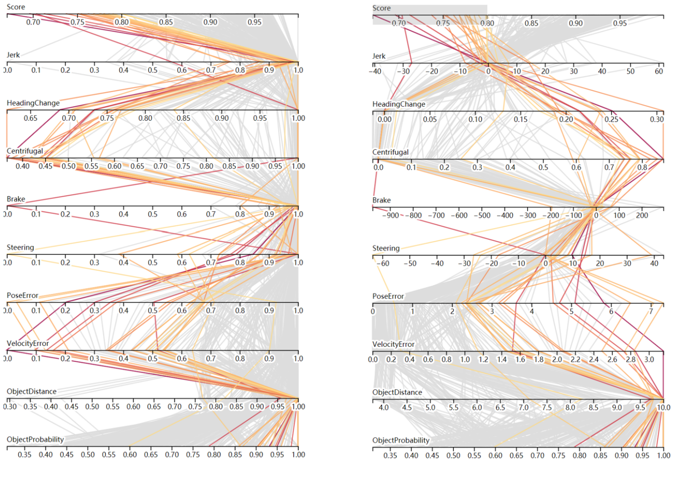
The spatial-temporal view (Fig. 1-a) shows the information of the environment and other traffic participants in an autonomous driving process. Its design goal is to provide the overall understanding of the spatial-temporal scenarios with animation. Users can examine the behaviors of the vehicle in this view when they try to identify the low scores at a specific time period. The green lines in the spatial-temporal view show the predicted movements of the obstacles. The thicker green line indicates the planned trajectories of ego-vehicle. The fences reflect planning decisions made by the planning module. Each type of decision is presented in different color. A red fence means that the detected obstacle stops moving forward while a purple fence means that the ego-vehicle is yielding. With the fence and predicted route visualization, users can easily understand the planned and perceived behaviors of the ego-vehicle and traffic participants. By dragging the progress bar in the spatial-temporal, user can review the state data and score data at any point of time.

To help users understand both the evaluation result (score) and relevant contributing factors, we use parallel coordinates to visualize the factor values (Fig1-d.3). Thus, it is consistent that users can investigate the score distribution and the factor value distribution in the same visual form. The top of the chart is still the score axis, and the factor axes below show the actual values of individual factor at each time point. Among them, Jerk, Headingchange, Centrifugal belong to the comfort module; Brake and Steering are about the control module; PoseError and VelocityError concern the planning module; ObjectDistance represents the perception module; and Probability is related to the prediction module. Thus, users not only understand how well a module performs, but also know how individual factors in each module contribute to it.

In addition, the state component visualizes the standard attributes of the autonomous vehicle and the obstacles during the autonomous driving process (Fig. 1-e), including the speed, acceleration, and wheel steering of the autonomous vehicle. This can help users directly understand the autonomous driving process. The obstacle attribute view (Fig. 1-e2) displays the information of the obstacles surrounding the vehicle during the autonomous driving process. It visualizes the type and priority strategy information of obstacles in the form of a stacked histogram.

4.2.3 Interactive Visual Evaluation

After the system evaluation is completed, users can review an autonomous driving process by clicking a button to animate the process, or by dragging the timeline to watch the animation. Users can also observe the time view below, drag the timeline, and explore the factor distribution of the five modules. After that, users can select and observe the factors in the view of parallel coordinates, find the factors with low scores, and examine the characteristics of the assessed factor values to understand the causes of failure. Finally, users can do interactive filtering and exploration on the parallel coordinates view, swipe a segment of the evaluation factors of interest to take values, and highlight the distribution of each evaluation factor at the corresponding time. Fig. 3 shows a scenario in which a user brushes the objects with scores less than 0.8 on the score axis of the factor score chart, view the evaluation results of the factors that influence such scores, switch to a view to display the same objects on the factor value chart, and examine the values of the factors that influence the evaluation.



Evaluation Score

Factor Value

Fig. 3: Parallel Coordinate Brush: the evaluation scores and the factor values of each factor at the same time point

4.2.4 Interactively Customized Ranking

The weights of all evaluation factors are equally distributed in the initial state. Because module developers with different background and interest may view different factors with different priorities. Users can drag the bars on the factor ranking visualization (R5, Fig. 1-d.2) to reorder their own priority. Dragging the factors that users consider important and moving the unimportant factors to the bottom can achieve the priority customization and update the evaluation modelling. In this way, we can get the new relative importance r between factors according to the relative positions of those bars, construct a new judgment matrix R, and calculate the new evaluation results according to AHP. As shown in Fig. 9, the left side is the parallel coordinate chart of the initial evaluation results. After reordering the factors' positions, users can get an updated evaluation scores showing on the right side.

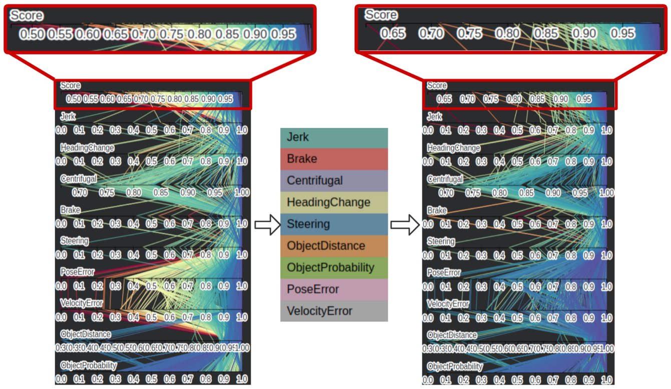


Fig. 4: Interactively customized ranking. The left figure shows the original evaluation scores with all the equal weight of modules. Users can drag the evaluation factor bars to reorder the importance of the factors, the new evaluation score results are generated on right figure.

4.3 Implementation

We use LGSVL Simulator [16] and Apollo [13] to implement our system. LGSVL Simulator is an end-to-end autonomous vehicle simulator that can be integrated with autonomous driving software and are compatible with various autonomous driving platforms. Apollo, an open-source autonomous driving platform that incorporates such modules as location, perception, control, prediction, and planning, has a component, Dreamview, to support the visualization of data from each module.

We use the Dreamview in Apollo as the basic framework to build our visual analysis system. Through the in-depth exploration of each module, we have refined a new set of visual evaluation methods. The visual evaluation system creates a perceptible environment for model developers through a series of real-time line graphs, radar view, stacked bar graphs, and parallel coordinates. It helps module developers better understand the performance of each module as the car moves. The system also supports user-driven definition of the importance of the evaluation index with interactive tools.

5 Case Study

We evaluate our system with three case studies, including an overall evaluation process on a car accident, a case to verify the specific autonomous driving module evaluation, and a case on the customized ranking in evaluation. All data involved is generated by LGSVL simulation.

5.1 Visual Evaluation for A Car Accident

This case is about a collision accident at an intersection without traffic lights where both vehicles make a left turn. This case used the BorregasAve map, with cars and pedestrians randomly generated by the simulator, to test the autonomous driving algorithms of Apollo. The collision, a severe accident, should lead to a low overall score at the time when the accident happens, which serves as a warning. We study whether the system would score the relevant modules reasonably because of the accident. We also evaluate the hundred-millisecond-level planning and prediction capabilities, which are essential metrics for assessing the capabilities of autonomous driving algorithms.

First, we get the total score and its distribution from the timeline view. As shown in Fig. 5, the total score is 89.06%, with a low score of 45% near the time 16:49:57.871. In the radar view, the scores of some modules are also very lower during this time period. Second, we continue to explore parallel coordinates and examine several factors with low scores, such as *HeadingChange, Centrifugal, Steering, PoseError, VelocityError, and ObjectDistance*. Third in the exploration, we replay the scene around the accident time and find that the vehicle under autonomous driving was turning left and colliding with another car on the right in the spatial temporal view. The distance between two vehicles was 0, so the perception module scores 0. During the collision, the vehicle was impacted and deflected to the extent that *HeadingChange, Steering, PoseError*, and other factors scores were affected. Apparently, the system correctly identifies the collision and gives a low score for driving status at the time when the accident happened.

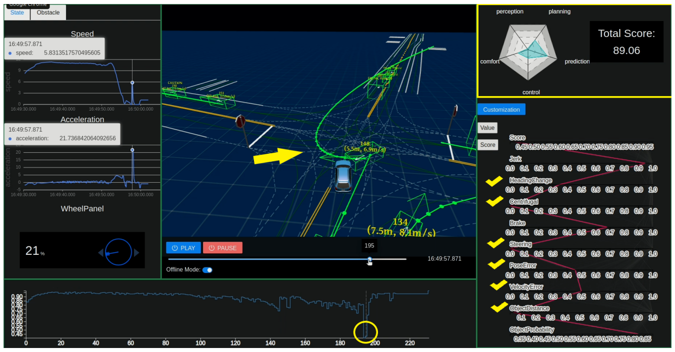
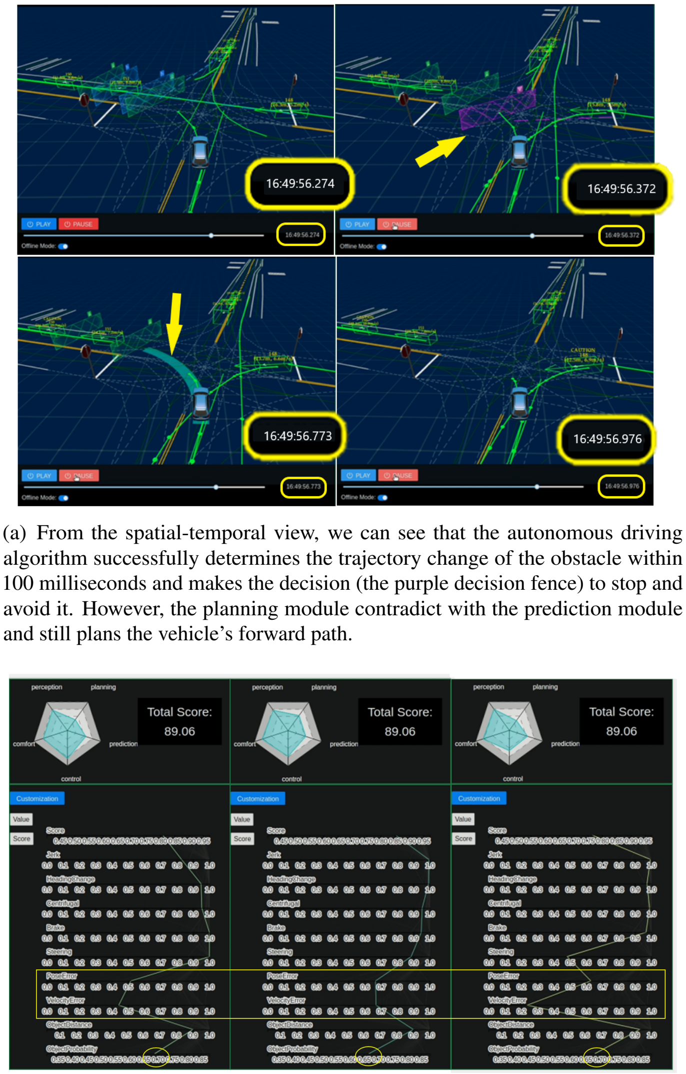


Fig. 5: A case of traffic accident. The yellow circle in the timeline view marks the lowest score. The spatial-temporal view shows the scene of the accident. The yellow ticks in parallel coordinates indicate the factors with poor performance at that moment.

Further exploring the visualized data, we can find some flaws in the autonomous driving behaviors. We explore the data around the time 16:49:57 (Fig. 6-a) to observe the scores of individual factors and behaviors of the vehicle. At 16:49:56.274, the Apollo autonomous driving algorithm had not yet predicted that the other vehicle was about to make a left turn. At 16:49:56.372, a purple decision fence appeared in front of the ego-vehicle, indicating a decision by the planning module to slow down the vehicle and to make space for the other vehicle on the right. However, after 0.4s at 16:49:56.773, the planning module changed its plan and decided to continue the movement. At 16:49:56.976, the autonomous driving system stopped the vehicle. From the radar view and parallel coordinates during this period (Fig. 6-b), we can see that the planning module and the prediction module have low scores. As shown in Fig. 6-b, Apollo can predict vehicle trajectories and plan vehicle routes within a time frame of a hundred milliseconds. Thus, the prediction module functions appropriately and can determine the trajectories of objects in complex road conditions. The prediction score is 0.65, indicating that there is still room for improvement in this module. If the algorithm could predict the trajectory of the other vehicle better, it might stop the ego-vehicle earlier, outside the course of the other vehicle. This case suggests that algorithm developers need to improve algorithms to ensure that sudden changes of the trajectories of objects around the vehicle do not create a new risk of collision within a time frame of a hundred milliseconds. The planning module, on the other hand, performed poorly during this period of time. After the prediction module predicted the trajectory of the other vehicle, the planning module still planned to continue the movement at 16:49:56.773. This error also reflects the inconsistency of the prediction and planning module, which should be investigated further by the developers.

Overall, this case shows that the evaluation system correctly identifies the time of the collision accident and gives an appropriate score based on the performance of each module. Moreover, through interactive exploration, users can understand the correlation between the accident and the performance of individual modules.



(a) From the spatial-temporal view, we can see that the autonomous driving algorithm successfully determines the trajectory change of the obstacle within 100 milliseconds and makes the decision (the purple decision fence) to stop and avoid it. However, the planning module contradict with the prediction module and still plans the vehicle's forward path.

(b) Module Performance. Parallel coordinates show that the low scores of PoseError, VelocityError and ObjectProbability lead to the poor performance of the planning module and medium-poor performance of the prediction module.

Fig. 6: Exploring the correlation between the accident and module performance by reviewing the scene before the accident.

5.2 Autonomous Driving Module Evaluation

In this case, we aim at examining the evaluation functionality for each module in autonomous driving. The initial state is shown in Fig. 1.

From the radar view (Fig. 1-b), we can clearly see that the overall score of this autonomous driving data segment is 91.94%. The initial state of the radar view shows the overall evaluation of each module. The timeline view (Fig. 1-c) shows the evaluation results over time. We focus on the period when a vehicle in autonomous driving scores low when making turns. During this period, we learn from the radar view that the comfort and planning modules perform poorly. Seeing the score of each factor through parallel coordinates, we find the factors that cause the low scores of the above two modules: HeadingChange, PoseError, and VelocityError. To explore other factors with bad performance, we brush the low score range on the left part (< 80) of the parallel axes (Fig. 7). There are two main periods with low score, as shown in the timeline view, the second half of the turn and a short period before the turn. For these two scenarios, we explore each module.

For the comfort module, we analyze the evaluation results of HeadingChange through several steps. First, we use parallel coordinates to select the region where the score is below 0.6 and query the time points

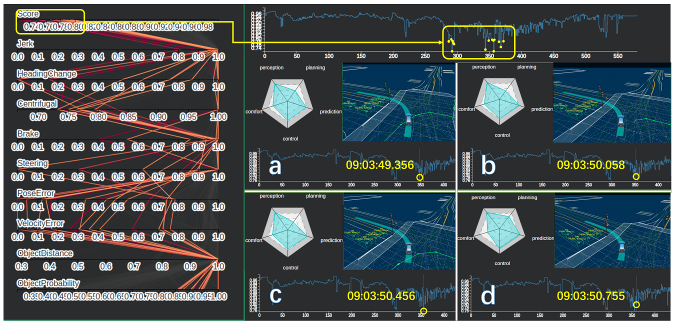


Fig. 7: Exploration of overall poor evaluation results. There are some poor evaluation results in the second half of the turn. a, b, c, and d shows time points with poor evaluation results in this turning process

of low scores. Second, we observe the distribution of low scores in the timeline view. These low scores are distributed in the second half of the turn. Third, we drag the progress bar of the spatial-temporal view to see the state information in specific scenarios. From the autonomous driving state data, we find that the WheelPanel changes frequently. Finally, we see from the radar view that the comfort module scores are low during the turning process. Observing the path of the vehicle, we find that it is slightly away from the intersection center during the turn. Apparently, the vehicle turned early and deviated from the optimal path (Fig. 8-a).

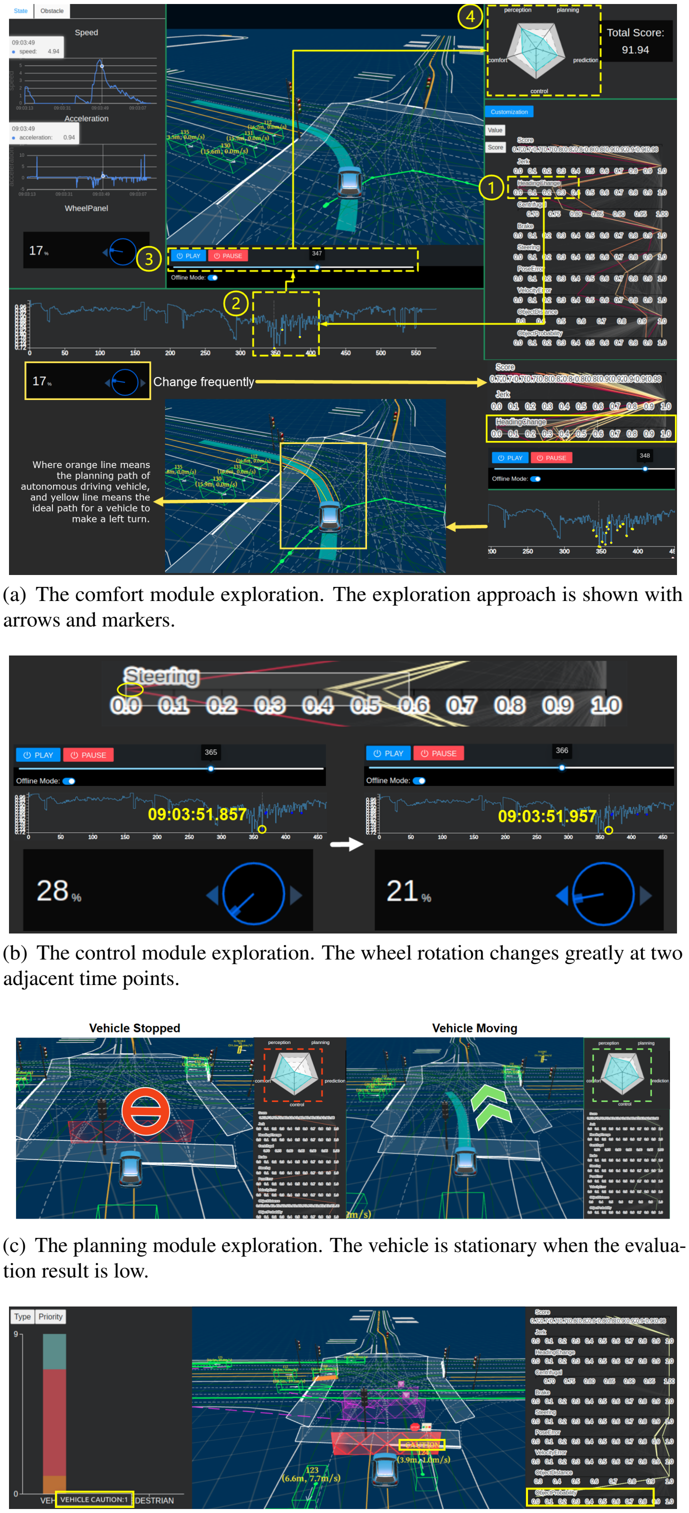
Similarly, we focus on the Steering factor of the control module. Brushing the area on the Steering axis where evaluation results are smaller than 0.6, we can see that the points mainly fall in the second half and the end of the vehicle turning process. Considering the fact that the vehicle deviates from the optimal path and the steering wheel rotation changes rapidly, we believe what is happening is that the control module is trying to bring the vehicle back on its planned course by making a series of adjustments to the wheel orientation (Fig. 8-b).

When analyzing the planning module, we find that PoseError and VelocityError tend to get low scores simultaneously. The scenario where both scores are below 0.4 occurs before the vehicle makes the turn. Dragging the progress bar and observing the scenario, we find that the planning score quickly picks up when the vehicle starts to move forward and turn. We speculate this is because the planning module needs information about the current state of the vehicle. When the vehicle moves, the information of its speed, acceleration, and other parameters are used as the reference for planning. However, when the vehicle is stationary, available information is insufficient for good planning. Therefore the accuracy of planning is relatively low (Fig. 8-c).

While exploring the ObjectProbability factor of the prediction module, we find some evaluation results with scores close to 0. We explore the corresponding scenarios by brushing the low-scoring area of this factor in parallel coordinates. The system highlights these scenarios in the timeline view. By dragging the progress bar to these time points and analyzing the obstacle view on the left, we find that the autonomous driving system identified an obstacle with a priority level of CAUTION. Through the spatial-temporal view, we can easily find that this obstacle is a car on the right side of the ego-vehicle. We consider that obstacles marked as CAUTION need special attention. Therefore, in our evaluation method, the evaluation of ObjectProbability relies heavily on the performance of the obstacles marked as CAUTION. At that moment, since the ObjectProbability value of the obstacle is 0, the system gives the score of the ObjectProbability factor as 0. In other words, the evaluation system considers that the prediction module has completely misjudged the trajectory of this obstacle (Fig. 8-d).

5.3 Customized Factors Ranking

From the user's point of view, the importance of individual modules varies in different situations. For object detection algorithm developers, they care more about the accuracy of their algorithms, so value the



Where orange line means the planning path of autonomous driving vehicle, and yellow line means the ideal path for a vehicle to make a left turn.

(a) The comfort module exploration. The exploration approach is shown with arrows and markers.

(b) The control module exploration. The wheel rotation changes greatly at two adjacent time points.

(c) The planning module exploration. The vehicle is stationary when the evaluation result is low.

(d) The prediction module exploration. There is one obstacle marked as CAUTION, and the accuracy of the prediction is 0

Fig. 8: Exploration of overall worse evaluation results.

performance of the perception and prediction modules most. However, for control algorithm developers, their interests are in the performance of the control module. When product managers evaluate an autonomous driving system, they probably focus on how comfortable the passengers of a vehicle or even the driver may feel in autonomous driving. Thus, evaluation methods should be flexible enough to accommodate different priorities by different stakeholders. In this case, we demonstrate how our system allows users to change the evaluation algorithm interactively.

We sort the factors according to their level of importance (Fig. 9). Assuming that users have greater interest in passenger comfort and vehicle planning route capability, they can elevate the comfort and planning modules in the list of sorted factors .

As shown in (Fig. 9), after the change of the order of factors, there is a clear tendency for the score to drop and then rise during the turn. Turning too fast can cause discomfort, so the scores of comfort factors drop accordingly. The values of Jerk, HeadingChange, and Centrifugal should be kept within a reasonable range, which algorithm developers should be concerned with.

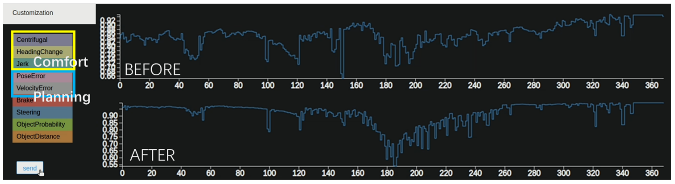


Fig. 9: Changing the order of factors leads to the change of the score line graph significantly.

On the other hand, the low planning score indicates that the planning algorithm performs poorly when the vehicle is turning, as reflected by the bias in estimating vehicle pose and velocity. In this case, the algorithm developer should consider how to improve the algorithm.

6 Expert Evaluation

In this research, we worked closely with domain experts in autonomous driving from a large car manufacturer. Five experts were involved in the collaboration from the definition of design requirements to the evaluation of the system.

To collect feedback on the system from experts, we organized a 90-minute workshop and invited three domain experts participated to evaluate system usability and to provide suggestions for improvement. The workshop started with a 15-minute tutorial, followed by the introduction of two cases that we presented in Sect. 5.1 and Sect. 5.2. The entire case study process took 20 minutes, including included a Q&A session. The experts then spent 40 minutes using the system for free exploration. After evaluating the cases and using the system, they filled out a questionnaire.

The questionnaire has 12 questions from three perspectives: 1) the correctness and intuitiveness of the visual evaluation method and interactive workflow; 2) the comprehension of the visualization designs in the system; 3) and suggestions for future improvement.

The overall feedback for our system and evaluation method is positive. In the workshop, all experts could follow the evaluation workflow and use our system to go through different cases. One expert believed that *"The system can visually demonstrate when, where, and how autonomous driving systems perform poorly."* They also confirmed that our case study is interesting and inspiring, as one expert said *"I'm convinced that this system can help experts quickly and efficiently identify the problems in autonomous driving."*

The experts agreed that the visualization tools provided by our system greatly help the evaluation process. For example, one expert praised the radar view by saying *"it helps users clearly identify which modules perform well or bad, supports good comparison, and also serves a good entrance for detailed analysis."* Most experts also liked the view of parallel coordinates and the interactive brushing function for linked view analysis. One expert said *"The interface design and interactive performance of the system are impressive. The correspondence between factor values and scores can be easily seen in parallel coordinates."*

7 Discussion and Conclusion

As autonomous driving systems are installed in more and more cars, it is very important to test their effectiveness and reliability before their deployment. The evaluation system of autonomous driving systems are needed to investigate the performances of various algorithms in different driving scenarios and to provide evidences for decisions on improvement and deployment.

There are some limitations in our method. The mathematical modeling approach we used, TOPSIS, is a method that amplifies the gaps between factors involved in evaluation. Although this treatment facilitates the identification of those subjects with poorer assessment results, it may overestimate the internal disparity of assessed factors. In the context of the evaluation of autonomous driving systems, even if driving data shows excellent performance at a certain time period, this method still tries to find some points in the process where the evaluation result is 0 on each factor. Consequently, autopilot data with good performance may still get poor assessment results somewhere. Although we consider adding a moderator γ when the extreme differences of the scoring data are minor, its values and the moderating criteria lack sound theoretical support.

In addition, although our visual evaluation system can identify most of the anomalous results in system testing, sometimes it misses the corresponding anomalies for the low scoring results identified by the system. This situation may be due to the fact that the system is sensitive to the changes of data inputs. Therefore, scores may fluctuate with the change of data.

Some weakness of our system is reflected at the data level. First, so far we only used a limited amount of simulation data without considering real-world data, only evaluated autopilot performances in normal weather conditions without including more challenging conditions (e.g., fog, rain), and only examined common autopilot behaviors without looking into more dangerous behaviors such as u-turns in busy traffics. To make our system more reliable and effective, we need to test it with more diverse driving scenarios that include broader driving behaviors based on both simulated and real-world data. In addition, we have not fully explored all driving data factors yet, and only used a few factors to evaluate the module performances. Considering the complexity of autonomous driving systems, we need to integrate more factors into the evaluation algorithm. Finally, although our evaluation methodology can be applied to simulation and live vehicle exercises, no road test evaluation have been conducted yet.

For the scalability of our system, to support batch analysis of multiple driving scenarios, we need to provide users with more flexibility in analytical procedure. Our current analytical procedure includes three main steps: 1) checking the distribution of the score timeline, 2) selecting module(s) with a low score to inspect the scores of related factors, and 3) examining the state showing factor values and spatial-temporal visualization. This procedure functions well for single-case analysis, but may be insufficient for batch analysis. It is needed to enrich the workflows and analytical procedures in our system.

We expect that our research can offer some new ideas for the evaluation of autonomous driving systems. In this work, we propose a visual evaluation system to assist the evaluation of an autonomous driving process, modules involved in autonomous driving, and the factors that contribute to the autonomous driving process. We constructed an evaluation model by combining mathematical methods of AHP and TOPSIS, and built a visualization system to support interactive evaluation of the results and state of autonomous driving data. The system can help the designers of autonomous driving systems to locate poorly performing moments, view the corresponding autonomous driving scenarios, and analyze and explain the reasons for low autonomous driving evaluation results. The outcomes of our system can help them better design and improve autonomous driving systems.

Acknowledgments

The authors wish to thank Lei Peng, Siqi Shen, Aolin Zhang, Weide Zhang and anonymous reviewers for their valuable suggestions. This work is supported by Shanghai Municipal Science and Technology Major Project 2018SHZDZX01 and ZJLab. This work is also supported by Shanghai Sailing Program No.21YF1402900 and Shanghai Science and Technology Commission (Grant No. 21ZR1403300).

References

[1] P. Koopman and M. Wagner. Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine*, 9(1):90–96, 2017.

[2] SAE International. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*.

[3] Contents and methods of field test capability assessment for automated vehicle. `http://jtgl.beijing.gov.cn/jgj/jgxx/gsgg/jttg/588465/683743/index.html`, 2018. last accessed on February 25, 2021.

[4] H. Huang, J. S. Albus, E. R. Messina, R. L. Wade, and R. W. English. Specifying autonomy levels for unmanned systems: interim report. In G. R. Gerhart, C. M. Shoemaker, and D. W. Gage, editors, *Unmanned Ground Vehicle Technology VI*, volume 5422, pages 386–397. International Society for Optics and Photonics, SPIE, 2004.

[5] K. Meng, Y. Zhao, L. Gao, and H. Tan. *Evaluation of the Intelligent Behaviors of Unmanned Ground Vehicles Based on Information Theory*, pages 410–419.

[6] F. Dong, Y. Zhao, and L. Gao. Application of gray correlation and improved ahp to evaluation on intelligent u-turn behavior of unmanned vehicles. In *2015 8th International Symposium on Computational Intelligence and Design (ISCID)*, volume 1, pages 25–29, 2015.

[7] J. Chen, P. Zhao, H. Liang, and T. Mei. A multiple attribute-based decision making model for autonomous vehicle in urban environment. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 480–485, 2014.

[8] Apollo dreamview. `https://github.com/ApolloAuto/apollo/tree/266afbf68d83fa6fac7a812ff8a950223f5ab2c0/modules/dreamview`. last accessed on June 30, 2021.

[9] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. Howard, S. Kolski, A. Kelly, M. Likhachev, M. Mcnaughton, N. Miller, and D. Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Robotics*, 25:425–466, 01 2008.

[10] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.

[11] T. Litman. Autonomous vehicle implementation predictions: Implications for transport planning. 2015.

[12] Autoware. `https://www.autoware.org/`. last accessed on March 2, 2021.

[13] Apollo. `https://apollo.auto/index_cn.html`. last accessed on February 25, 2021.

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. Carla: An open urban driving simulator, 2017.

[15] B. Osiński, P. Miłoś, A. Jakubowski, P. Ziecina, M. Martyniak, C. Galias, A. Breuer, S. Homoceanu, and H. Michalewski. Carla real traffic scenarios – novel training ground and benchmark for autonomous driving, 2020.

[16] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta, E. Agafonov, T. H. Kim, E. Sterner, K. Ushiroda, M. Reyes, D. Zelenkovsky, and S. Kim. Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020.

[17] Apollo dreamland. `https://bce.apollo.auto/`. last accessed on December 23, 2020.

[18] Y. Wang and J. Liu. Evaluation methods for the autonomy of unmanned systems. *Chinese Science Bulletin*, 57:3409–3418, 2012.

[19] Y. Zhao, X. Luo, X. Lin, H. Wang, X. Kui, F. Zhou, J. Wang, Y. Chen, and W. Chen. Visual analytics for electromagnetic situation awareness in radio monitoring and management. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):590–600, 2020.

[20] L. Ren, Y. Liu, X. Wang, J. Lü, and M. J. Deen. Cloud-edge based lightweight temporal convolutional networks for remaining useful life prediction in iiot. *IEEE Internet of Things Journal*, 2020.

[21] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, pages 1–34, 2020.

[22] L. Gou, L. Zou, N. Li, M. Hofmann, A. K. Shekar, A. Wendt, and L. Ren. Vatld: A visual analytics system to assess, understand and improve traffic light detection. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):261–271, 2021.

[23] B. Lars, S. Benno, S. Ulrich, and U. Christoph. Managing, modeling, and visualizing high-dimensional spatio-temporal data in an integrated system. *GeoInformatica*, 2:59–77, 03 1998.

[24] H. Guo, Z. Wang, B. Yu, H. Zhao, and X. Yuan. Tripvista: Triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection. In *2011 IEEE Pacific Visualization Symposium*, pages 163–170, 2011.

[25] S. Chen, X. Yuan, Z. Wang, C. Guo, J. Liang, Z. Wang, X. Zhang, and J. Zhang. Interactive visual discovering of movement patterns from sparsely sampled geo-tagged social media data. *IEEE transactions on visualization and computer graphics*, 22(1):270–279, 2015.

[26] A. Buja, D. Cook, and D. F. Swayne. Interactive high-dimensional data visualization. *Journal of Computational and Graphical Statistics*, 5(1):78–99, 1996.

[27] A. A. Goodman. Principles of high-dimensional data visualization in astronomy. 2012.

[28] A. M. Martinez and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[29] J. Yang, A. F. Frangi, J. Yang, D. Zhang, and Z. Jin. Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):230–244, 2005.

[30] Q. Liu, H. Lu, and S. Ma. Improving kernel fisher discriminant analysis for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):42–49, 2004.

[31] X. Geng, D. Zhan, and Z. Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, 2005.

[32] K. P. Singh, R. Bhai, V. Mishra, P. Nagar, and J. Kasinayal. Localization in wireless sensor network using lle-isomap algorithm. In *TENCON 2017 - 2017 IEEE Region 10 Conference*, pages 393–397, 2017.

[33] N. Andrienko, G. Andrienko, and P. Galalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages Computing*, 14(6):503–541, 2003.

[34] G. Andrienko, N. Andrienko, G. Fuchs, and J. Wood. Revealing patterns and trends of mass mobility through spatial and temporal abstraction of origin-destination movement data. *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2120–2136, 2017.

[35] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, 2013.

[36] S. Chen, G. Andrienko, N. Andrienko, C. Doulkeridis, and A. Koumparos. Contextualized Analysis of Movement Events. In Tatiana von Landesberger and Cagatay Turkay, editors, *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2019.

[37] T. L. Saaty. A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234–281, 1977.

[38] E. Triantaphyllou. *Multi-Criteria Decision Making Methods*, pages 5–21. Springer US, Boston, MA, 2000.