Journal of Intelligent and Connected Vehicles
2023, **6**(4): 264–274
https://doi.org/10.26599/JICV.2023.9210023

Research Article

SceGAN: A method for generating autonomous vehicle cut-in scenarios on highways based on deep learning

Lan Yang, Jiaqi Yuan✉, Xiangmo Zhao, Shan Fang, Zeyu He, Jiahao Zhan, Zhiqiang Hu, Xia Li

School of Information Engineering, Chang'an University, Xi'an 710064, China

Received: September 8, 2023; Revised: September 14, 2023; Accepted: September 18, 2023

ABSTRACT: With the increasing level of automation of autonomous vehicles, it is important to conduct comprehensive and extensive testing before releasing autonomous vehicles into the market. Traditional public road and closed-field testing failed to meet the requirements of high testing efficiency and scenario coverage. Therefore, scenario-based autonomous vehicle simulation testing has emerged. Many scenarios form the basis of simulation testing. Generating additional scenarios from an existing scenario library is a significant problem. Taking the scenarios of a proceeding vehicle cutting into an adjacent lane on highways as an example, based on an autoencoder and a generative adversarial network (GAN), a method that combines Transformer to capture the features of a long-time series, called SceGAN, is proposed to model and generate scenarios of autonomous vehicles on highways. An evaluation system is established to analyze the reliability of SceGAN using discriminative and predictive scores and further evaluate the effect of scenario generation in terms of similarity and coverage. Experiments showed that compared with TimeGAN and AEGAN, SceGAN is superior in data fidelity and availability, and their similarity increased by 27.22% and 21.39%, respectively. The coverage increased from 79.84% to 93.98% as generated scenarios increased from 2,547 to 50,000, indicating that the proposed method has a strong generalization capability for generating multiple trajectories, providing a basis for generating test scenarios and promoting autonomous vehicle testing.

KEYWORDS: scenario generation, autonomous vehicles testing, cut-in, transformer, generative adversarial network (GAN)

1 Introduction

Intelligent transportation system (ITS) with its powerful communications, networks, and sensor devices, makes human life convenient, especially in path planning, ride demand forecasting, intelligent perception, etc. (Liu et al., 2022a, 2022b, 2023; Zong et al., 2022b). Before releasing autonomous vehicles into the market, they must undergo extensive testing to ensure their safety. Traditional public road testing is time-consuming and unsafe, whereas closed-field testing has a limited scope, and exhaustive testing is difficult to achieve. Scenario-based simulation testing has become the most effective supplementary solution in autonomous vehicles' the development and validation stages, accounting for more than 90% of autonomous vehicle testing. Approximately 10^8 test scenarios are required to simulate driving in the simulation environment. Therefore, producing repeatable scenarios under conditions similar to those in a real environment is crucial for simulation testing.

A scenario is a dynamic process in which multiple traffic participants interact in a certain traffic environment over time (Ulbrich et al., 2015). One of the important sources of scenarios is to extract data from natural driving datasets and generate more data using scenario-generation methods to build diverse and complex scenario libraries. These scenarios can be inserted into simulation platforms to validate the safety of the autonomous vehicles.

Current methods for scenario generation can be divided into random sampling, parameter combination, and machine learning. Random sampling generates high-authenticity data but is less suitable for large-scale scenario generation. Parameter combination considers various elements that display traffic scenarios but is prone to generating scenarios that are unreasonable and require extremely high manual intervention or robust post-processing. The machine-learning-based approach can automatically generate a substantial number of test scenarios, but it requires a significant volume of original scenario data for model training. Although different methods have their advantages and disadvantages, machine learning-based methods remain the most mainstream methods; especially the generation of time-series data by GAN, is a key research topic for promoting the development of ITS and solving the problem of insufficient scenarios (Lin et al., 2023). Cut-in scenarios, which are among the most risk-prone pre-collision scenarios, must be addressed urgently for safety testing.

However, there are certain challenges in this research. First, the generation of long-time trajectory under cut-in scenario is easy to lead to a zigzag problem. Moreover, the generation process is often uncontrolled. Second, to fully verify the safety of autonomous vehicles, multiple scenarios must be generated. However, generating more data from a small amount may result in a poor generalization of the time-series model, making scenario generation less scalable. Therefore, it is necessary to design a model that can generate smooth trajectories and has a great

✉ Corresponding author.
E-mail: yuanjiaqi312@163.com

generalization ability.

Based on existing research, this study's primary contributions are as follows:

1) In the vehicle kinematics model, we considered several variables, such as initial position and instantaneous speed, to extract the highway cut-in behaviors.

2) We introduced a scenario generation method called SceGAN, which combines an autoencoder, generative adversarial network (GAN), and Transformer, to enable the automatic generation of virtual test scenarios.

3) We established an evaluation system to prove the reliability of the SceGAN and the similarity and coverage of the generated scenarios.

2 Related works

As mentioned in Section 1, there are three main methods for scenario generation research. Random sampling methods generate typical or critical scenarios using random or importance sampling. Zhu et al. (2022) used Markov chain and Monte Carlo sampling to generate many lane-changing scenarios with high coverage and similarity; however, the smoothness of the generated trajectories was poor. Xu et al. (2018) combined a genetic algorithm and importance sampling to generate risky cut-in scenarios based on the driving characteristics in China. Yang et al. (2021) used Gibbs sampling to generate 20,000 cut-in scenarios and evaluated comfort. Zhou et al. (2021) combined the Monte Carlo method with importance sampling to generate test cases for cut-in scenarios, thereby increasing the number and proportion of high-risk scenarios. Feng et al. (2022) used the Latin hypercube sampling method to initialize particles when searching for critical scenarios. Sampling methods must consider the number and location of the sampling points, which may lead to limitations and insufficiencies in the generated scenarios.

The parameter combination method combines scenario elements by designing rules and often uses an ontology to analyze scenarios. Shu et al. (2019) considered the possible driving behaviors of traffic participants interacting with autonomous vehicles and generates many combined scenario groups. Duan et al. (2022) added a test matrix to a combination testing method, considering the scenario complexity and testing cost. Subsequently, they implemented scenario generation for a lane departure warning system. Li et al. (2020) used ontology to describe the environment of autonomous vehicles and transformed it into testing scenarios using a combination testing method. Hu et al. (2022) generated scenarios by combining parameters from classifications based on road types and driving tasks and designed a constraint set algorithm to solve the unreasonable scenarios problem. Rocklage et al. (2017) used a combination of interaction testing and trajectory planners to generate an efficient test set for regression testing of autonomous vehicle systems. However, despite its high controllability and predictability of diversity, combination-based methods are susceptible to artificial limitations.

Machine learning methods can automate the learning process of intrinsic patterns and features through large iterations of the training data, which can be used to build high-fidelity libraries in diverse test scenarios owing to its strong generalization capability. Jenkins et al. (2018) considered the traffic signal status and used a recurrent neural network (RNN) to generate accident scenarios. Tan et al. (2021) used a neural autoregressive model to generate intersection scenarios considering pedestrians and various vehicles types. Demetriou et al. (2023) generated cut-in scenarios using two types of GAN. The scenarios generated by recurrent condition GAN have poor performance, while autoencoder GAN performs well. To enhance the interpretability of the deep-learning-based scenario generation method, Krajewski et al. (2018a) used Info-GAN and beta-VAE to generate lane-changing trajectories from latent variables. Hoseini et al. (2021) proposed a recurrent autoencoder GAN to generate additional cut-in scenarios for data augmentation during clustering. Feng et al. (2021) presented a deep knowledge of reinforcement learning methods by proposing an adaptive scenario generation framework to construct libraries for various critical test scenarios. Moreover, they used augmented reality environments that combined background vehicles, road infrastructure, and autonomous vehicles to test tracks on highways and in urban areas (Feng et al., 2023). The machine-learning-based approach has a high degree of automation in the process but requires large data to support the training of the model.

To realize the automatic generation of test scenarios for cut-in behavior, this study proposes SceGAN, which combines Transformer, autoencoder, and GAN, then generates many cut-in scenarios to improve coverage while ensuring high similarity.

3 Methodology

In this section, we mainly describe the definition and representation of cut-in scenarios, the generation process in SceGAN and the concrete network structure of SceGAN in detail.

3.1 Scenario representation

As shown in Fig. 1, a cut-in scenario is defined as a situation on a highway where a lane-changing vehicle (i.e., a cut-in vehicle, denoted as CV hereafter) interacts with an autonomous vehicle (i.e., an ego vehicle, denoted as EV hereafter) driving behind it in the target lane. Let $S = \{f, a\}$ be the instance of an input scenario. Because the scenario positions, accelerations, and other characteristics change over time, these elements can be considered as features $\boldsymbol{f} = \{f_1, f_2, \ldots, f_T\} \in \mathbb{R}^{T \times n}$, where T represents the total time step , and $\boldsymbol{f}_t = \{f_t^1, \ldots, f_t^n\} \in \mathbb{R}^n$ at time step t, which is a vector with dimension n. Elements, such as the inherent properties of the vehicles that do not change with time, can be represented as $a \in \mathbb{R}^k$, where k is the dimension of attributes.

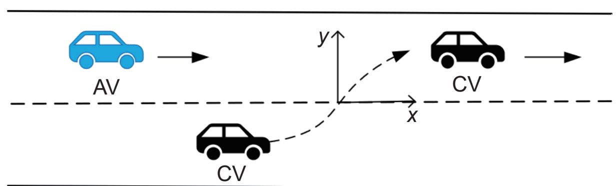


Fig. 1 Cut-in scenario on highways.

3.2 Architecture

Fig. 2 depicts the architecture of SceGAN, which is based on RTSGAN (Pei et al., 2021). The automated generation of cut-in scenarios is accomplished using an autoencoder and a GAN combined with Transformer modules. Before training and generation, we normalize all data X from the extracted cut-in scenarios to X_{Norm}.

$$X_{\text{Norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{1}$$

where $X_{\min}$ and $X_{\max}$ are the minimum value and maximum value of X.

https://doi.org/10.26599/JICV.2023.9210023

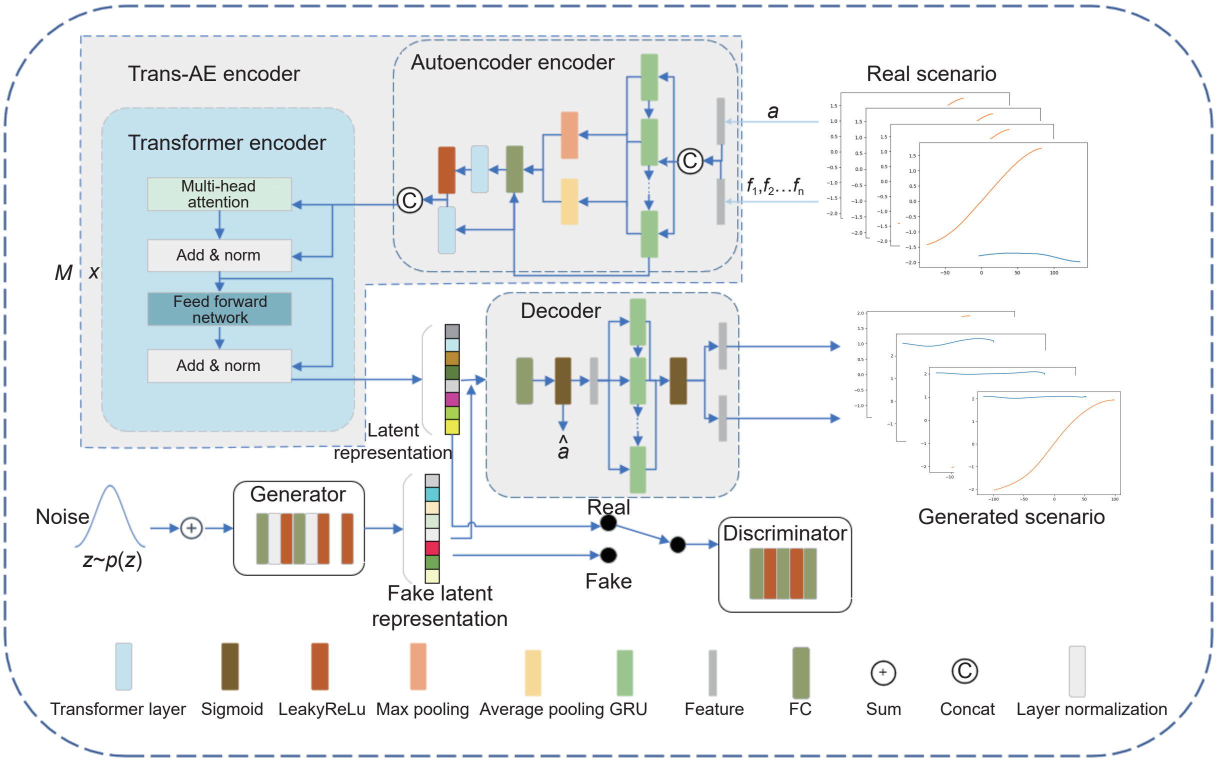


Fig. 2 Architecture of SceGAN.

The trans-AE encoder establishes a mapping relationship between time-series instances (i.e., scenario data) and fixed-dimensional latent vectors, and its module consists of an autoencoder (AE-encoder) and a Transformer encoder (TR-encoder) (Vaswani et al., 2017). The features and attributes are associated with complete scenarios $s_t = [f_t, a]$ at time step t in the AE-encoder, and the hidden states h_t^n are obtained using an N-layer gated recurrent unit (GRU) at layer n.

The model is expected to learn temporal information globally and generate trajectories with high continuity and smoothness. However, AE-encoder is not suitable for handling temporal information. A TR-encoder is introduced to improve the model and is used thrice, as shown in Fig. 2.

First, key information in is extracted using the maximum pooling layer and average pooling layer, and a fully connected (FC) neural network is used to aggregate the processed pooling results.

$$u = \mathrm{FC}\left(\mathrm{maxpool}\left(h_t^N\right), \mathrm{avgpool}\left(h_t^N\right)\right), t \in (1, T) \tag{2}$$

where h_t^N is hidden states from the last layer of GRUs at time step t.

The TR-encoder module contains multiple encoder layers. Each layer receives hidden information from the output of the last layer and consists of multi-head attention and a fully connected feedforward neural network, which can encode long-term sequence data with strong temporal modeling capability. The self-attention is calculated as Eq. (3):

$$\mathrm{Attention}(q, k, v) = \mathrm{Softmax}\left(\frac{qk^T}{\sqrt{d_k}}\right) v \tag{3}$$

where q, k, and v are the query, key, and value matrices, respectively, obtained by the linear transformation of the received input u. d_k is the dimension of the key matrix. The softmax activation function is used to calculate the attention weight.

Multi-head attention is a combination of multiple self-attentions, capturing the contextual information at other locations in the time series and weighing the information to obtain a richer representation. Let u_t be multi-head attention:

$$u_t = \mathrm{concat}(head_1, \ldots, head_L) W^o \tag{4}$$

where t denotes the time step of the input sequence, W^O is the output weight matrix, and $head_i$ is the i-th head:

$$head_i = \mathrm{Attention}\left(q_t \cdot W_i^q, k_t \cdot W_i^k, v_t \cdot W_i^v\right), i \in (1, L) \tag{5}$$

where W_i^q, W_i^k, and W_i^v are weight matrices for linear transformations.

The inputs u and outputs u_t of the multi-head attention mechanism are added to focus on the information at the current time step. Layer normalization (LN) is used to accelerate the convergence of the network.

$$x = \mathrm{LN}\left(u + u_t\right) \tag{6}$$

The feed-forward network (FFN) converts the encoded results into a two-layer fully connected layer using a nonlinear transformation. Moreover, the add and layer normalization operations are performed again to obtain the latent representation of the entire encoder.

$$\mathrm{FFN}(x) = \max\left(0, W_1 x + b_1\right) W_2 + b_2 \tag{7}$$

$$y_1 = \mathrm{LN}\left(x + \mathrm{FFN}(x)\right) \tag{8}$$

$$y_1 = \text{TR-encoder}(u) \tag{9}$$

where W_1, b_1, and W_2, b_2 are learnable weights and biases, and y_1 is the final latent representation of the first TR-encoder. From Eq. (3) to (8), the encoding process of the TR-encoder is completed, which is summed up in Eq. (9).

Second, the TR-encoder is applied to the last layer of GRUs h_T^N, and y_1 is concatenated to the hidden layer:

$$h_T^N = \text{TR-encoder}\left(h_T^N\right) \quad (10)$$

$$y_2 = \left[y_1, h_T^N\right] \quad (11)$$

where y_2 is new latent representation of the hidden layer, and h_T^N is hidden state at last time step.

Finally, to ensure that both the position and speed information are relatively compressed completely, a TR-encoder is used to extract a rich feature representation.

$$r = \text{TR-encoder}\left(y_2\right) \quad (12)$$

where output r represents a latent representation of the entire encoding phase.

The decoding process is the inverse of the AE-encoder process. It decomposes the time series from the generated latent vectors with a distribution like the original features. We first reconstruct static attributes from the latent representation, and dynamic features based on these values. Attribute reconstruction is primarily realized through a fully connected layer. Feature reconstruction is performed using the GRU, which is derived from the last hidden layer. The reconstruction loss during encoding and decoding is represented as L_{re}, as shown in Eq. (13):

$$L_{\text{re}} = \frac{d_a}{d_f + d_a}\text{MSELoss}\left(\widehat{a}, a\right) + \frac{d_f}{d_f + d_a}\text{MSELoss}\left(\widehat{f}, f\right) \quad (13)$$

where $\widehat{a}$ and $\widehat{f}$ are reconstructed attributes and features.

After training the encoder and decoder modules, Wasserstein GAN (WGAN) (Gulrajani et al., 2017) is trained to generate vectors and the generator is trained by minimizing the 1-Wasserstein distance between the distributions of the real and synthetic data. A 1-Lipschitz discriminator was used to optimize the objective function of the WGAN.

$$\min_G \max_D E_{r\sim\text{encoder}(f,a)}\left[D\left(r\right)\right] - E_{z\sim p(z)}\left[D\left(G\left(z\right)\right)\right] \quad (14)$$

where z represents the noise. G and D are the generator and discriminator, respectively. Using the WGAN and the 1-Lipschitz discriminator, the GAN can be optimized more consistently during training to generate more realistic data.

4 Evaluation metrics

The evaluation system included reliability, similarity, and coverage. Reliability is primarily used to compare the data fidelity and availability of the method. Similarity and coverage are evaluated for the generated scenarios when generating an equally proportional number of scenarios and multiples of scenarios, respectively. In addition, considering other factors in the scenario elements such as roads and weather, complexity is also a significant evaluation method (Li et al., 2022).

The current metrics for time-series data generation are mainly fidelity and availability, as proposed by TimeGAN (Yoon et al., 2019), and are expressed by discriminative and predictive scores. The discriminative score is used to distinguish the real dataset from the generated dataset by training a two-layer long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997), and a lower score indicates higher fidelity of the generated dataset. The predictive score predicts the state of the time series at the next time step by training the two-layer LSTM under the “train on synthetic, test on real” (TSTR) setting. The predictive score characterizes the continuity and smoothness of the sequences to a certain extent and describes the feasibility of the trajectory data for prediction.

Similarity analyzes the distance between generated and original scenarios from the perspectives of data distribution and visualization when generating proportional data. The Jensen–Shannon divergence (JS divergence) is suitable for comparing the distributions of two datasets of the same shape. This is a variation in the Kullback–Leibler divergence (KL divergence). Because JS divergence solves the problem of asymmetric KL divergence, it is more universal (Barz et al., 2019).

$$D_{\text{JS}}\left(\widehat{S}\middle\|S\right) = \frac{1}{2}D_{\text{KL}}\left(\widehat{S}\middle\|\frac{\widehat{S}+S}{2}\right) + \frac{1}{2}D_{\text{KL}}\left(S\middle\|\frac{\widehat{S}+S}{2}\right) \quad (15)$$

$$D_{\text{KL}}\left(\widehat{S}\middle\|S\right) = \sum\nolimits_{i=1}^{n}\widehat{S}\log\frac{\widehat{S}}{S} \quad (16)$$

where $\widehat{S}$ and S denote the generated and original data, respectively.

The coverage metric characterizes the ability of SceGAN to learn the data distribution and generalize it. Dynamic time warping (DTW) is used to evaluate the generated scenarios (Demetriou et al., 2023). DTW uses dynamic programming to calculate coverage and matching. However, this method requires traversing the data to find scenarios with the closest DTW distance, which has a time complexity of $O\left(n^2\right)$.

Intersection over union (IOU) (Yu et al., 2016) is an accuracy measure for detecting corresponding objects in a specific dataset in the field of target detection. It is calculated by computing the ratio of the areas of the intersection and union sets of the two boxes. This study introduces the IOU as a coverage evaluation metric by first downscaling each scenario into two-dimensional data points using t-distributed stochastic neighbor embedding (t-SNE) (Maaten and Hinton, 2008) and then calculating the IOU.

$$\text{IOU} = \frac{\text{sum}\left(p_i \text{ in } P \text{ and } p_i \text{ in } \widehat{P}\right)}{\text{sum}\left(p_i \text{ in } P \text{ or } p_i \text{ in } \widehat{P}\right)}(i = 1, 2, \ldots, n) \quad (17)$$

where P and $\widehat{P}$ are point sets of downscaled original and generated scenarios, respectively.

To verify that the generated scenarios have significant coverage in each risk interval, a risk assessment method needs to be introduced. The risk of a cut-in scenario often arises from the short car-following process of neighboring vehicles after changing lanes (Zong et al., 2022a). The standard collision risk (SCR) can be used when potential risk of braking behavior is considered (He et al., 2023). Since the deceleration of lane-changing vehicles is not obvious, this study refers to the grading method (Essa and Sayed, 2019) for time to collision (TTC) to measure the risk of the scenarios. Accordingly, we establish four risk levels as in Eq. (18):

$$\text{risk level} = \begin{cases} \text{highrisk}, & \dfrac{1}{t_{\text{TTC}}} \geq 1 \\ \text{mediumrisk}, & 0.5 \leq \dfrac{1}{t_{\text{TTC}}} < 1 \\ \text{low risk}, & 0 \leq \dfrac{1}{t_{\text{TTC}}} < 0.5 \\ \text{safe}, & \text{others} \end{cases} \quad (18)$$

5 Experiments and results

The experiments were performed on an equipment with a CPU (i9-12900KF), GPU (RTX 3090), and 64 GB memory. The baseline methods for this study were TimeGAN (Yoon et al.,

2019) and AEGAN (Demetriou et al., 2023), and the data were 2,547 lane-change scenarios explicitly extracted from highD (Krajewski et al., 2018b). We set all the extracted lane-changing positions as the origin, intercepted the 6 s before and after the lane-change points, and obtained the data with a uniform length of 150-time steps to better visualize these trajectories. Table 1 lists the experimental parameters and their corresponding values. The two learning rates originated from the autoencoder and WGAN, whereas the two batch sizes were determined using the autoencoder and WGAN. Under our experimental conditions and parameter settings, the training and generation of scenarios typically require approximately 1.45 h.

Table 1 Parameters and values

Parameter	Value
Epoch	1,000
Learning rate1	0.001
Learning rate2	0.0001
Gradient penalty term	10
Iteration	15,000
Batch size1	128
Batch size2	256
Hidden dim	75
Noise dim	300
Dropout	0.2
Head	4
Layer	2

5.1 Reliability

To verify the reliability of SceGAN in terms of data generation fidelity and availability, discriminative and predictive scores were evaluated five times and their average values were recorded. As Table 2 shows, SceGAN had the lowest scores. The fidelity of the data generated by SceGAN increased by 72.62% and 55.30%, and the availability increased by 52.58% and 10.64%, respectively.

Table 2 Methods performance and ablation experiment (lower the better)

Method	Discriminative	Predictive
AEGAN	0.2076	0.0329
1st TR-encoder only	0.1675	0.0321
2nd TR-encoder only	0.1924	0.0296
3rd TR-encoder only	0.1125	0.0299
1st and 2nd TR-encoders	0.1049	0.0292
1st and 3rd TR-encoders	0.0970	0.0297
2nd and 3rd TR-encoders	0.0983	0.0295
SceGAN	**0.0928**	**0.0294**
TimeGAN	0.3390	0.0620

In reality, the training time increases with the number of modules. Therefore, if there is a strong emphasis on minimizing the training time, one may consider adding only the third TR-encoder, as it exhibits the lowest discriminative score in situations when there is minimal variation in predictive scores during ablation experiments.

5.2 Similarity

Fig. 3 shows the distributions of the trajectory and speed values of the generated and original scenarios and their similarities. Evidently, the TimeGAN results are unevenly distributed. The scenarios generated by AEGAN approximately captured the data distribution of the original scenarios; however, they are not as close to the results of SceGAN. Quantitatively, the average JS divergence from Table 3 shows that the similarity improved by 27.22% and 21.39%, compared to TimeGAN and AEGAN, respectively.

Figs. 4 and 5 show the speed–time curves and trajectories, respectively. Distinct cut-in behavior characteristics can be observed in the generated scenarios. The EV is behind the target lane of the CV, and its driving process has a weak regularity, as shown in Fig. 4a. However, SceGAN learned its general characteristics well and generated most of the speed values at [−0.4, 0.4].

Although SceGAN works well overall, some detailed problems remain, and the generated scenarios tend to be intermediate. For example, there are scenarios with longitudinal velocities close to 50, as shown in Fig. 4a, whereas the curves in Fig. 4b tend to reach approximately 42. The scenarios generated by SceGAN showed smaller fluctuations in the lateral speed of the CV compared to the original scenarios.

As Fig. 5 shows, the trajectories generated by TimeGAN are messy, whereas those generated by AEGAN and SceGAN are cleaner. However, the CV trajectories generated by AEGAN are not as smooth as those generated by the SceGAN. A few EV trajectories cross the point where the lateral position is zero, indicating that they cross the lane line, which does not correspond to the actual situation.

5.3 Coverage

SceGAN generated a set of scenarios to test the coverage, as presented in Table 4. With an increase in the number of generated scenarios, the proportion of risky scenarios increased to some extent but maintained a distribution similar to the original scenarios. This shows that SceGAN has a certain degree of scenario generalization ability.

The t-SNE visualization and IOU calculation results are shown in Fig. 6. The red dots represent the original 2,547 scenarios, whereas the blue dots represent different sets of generated scenarios (i.e., synthetic data). When the number of generated scenarios was 2,547, the coverage reached 0.7984, and SceGAN learned the data distribution. As the number of generated scenarios increased, the coverage gradually increased to 0.9398, and there was a relatively distinct outline, indicating that SceGAN effectively analyzed the data distribution and captured scenario boundaries.

6 Conclusions

In this study, we proposed a test scenario generation method called SceGAN, based on integrating an autoencoder, GAN, and Transformer, focusing on the cut-in behavior on highways. Multiple features such as vehicle speed and position were modeled and generated as time-series data. In terms of reliability, similarity, and coverage, the following conclusions were drawn:

1) The fidelity and availability of the data generated by SceGAN were verified by discriminative and predictive scores, which were 55.30%, 72.62%, and 10.64%, 52.58% higher than those of TimeGAN and AEGAN, respectively.

2) By visualizing the data distribution and scenario curves, the scenarios generated by SceGAN were more similar to the original scenarios. According to the JS divergence, the similarity increased

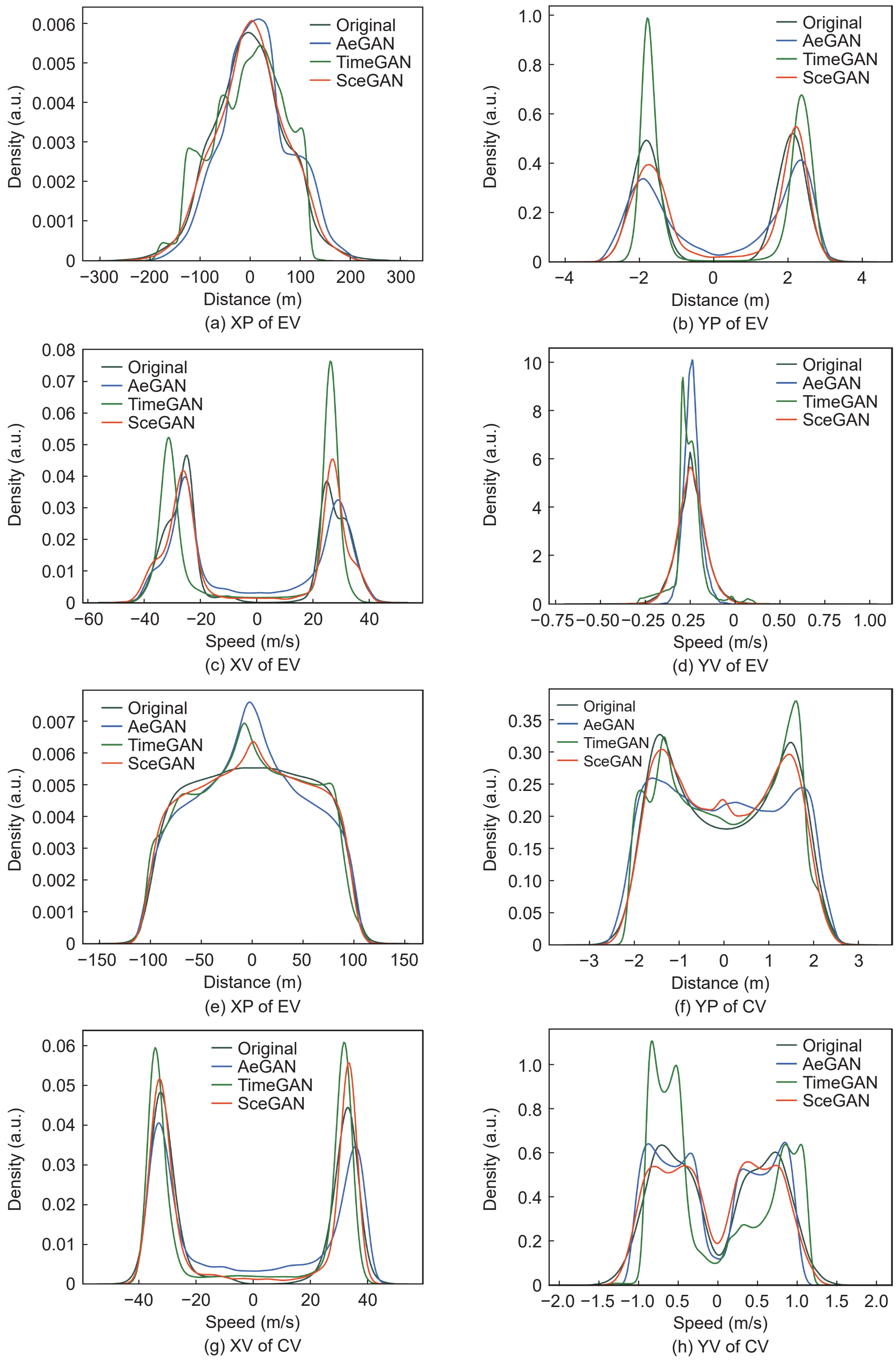
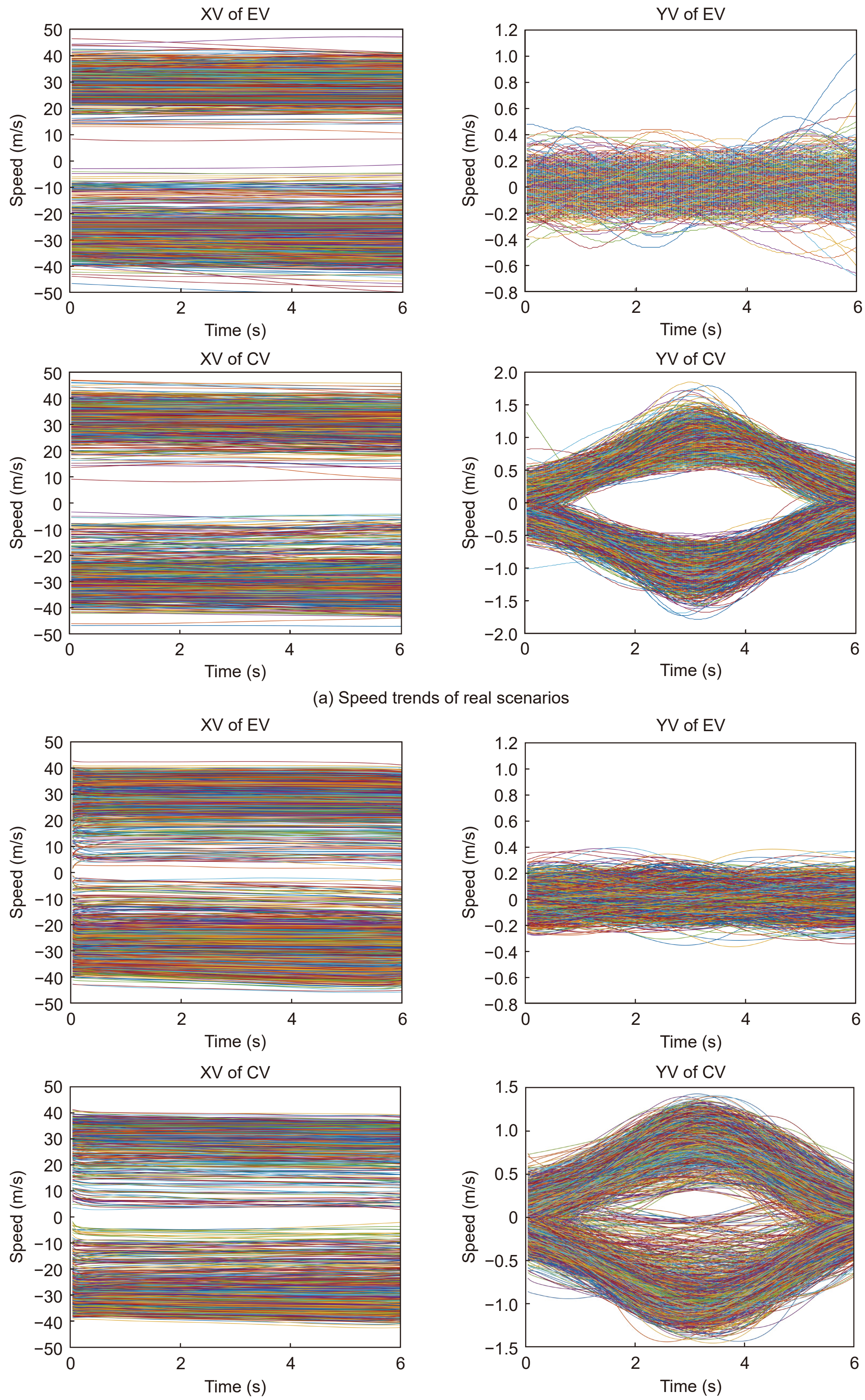


Fig. 3 Estimation of the density of each variable for the original and generated scenarios (XP: longitudinal position; YP: lateral position; XV: longitudinal speed; YV: lateral speed).

Table 3 Js divergence of ego vehicles and cut-in vehicles

Method	XP of EV	YP of EV	XP of CV	YP of CV	Average
TimeGAN	0.0227	0.0841	0.0884	0.0639	0.06478
AEGAN	0.0207	0.1151	0.0519	0.0522	0.05998
SceGAN	0.0163	0.0782	0.0471	0.0470	0.04715

清華大学出版社 Tsinghua University Press | IEEE



(a) Speed trends of real scenarios

(b) Speed trends of generated scenarios

Fig. 4 Longitudinal and lateral speed trends of ego vehicles and cut-in vehicles.

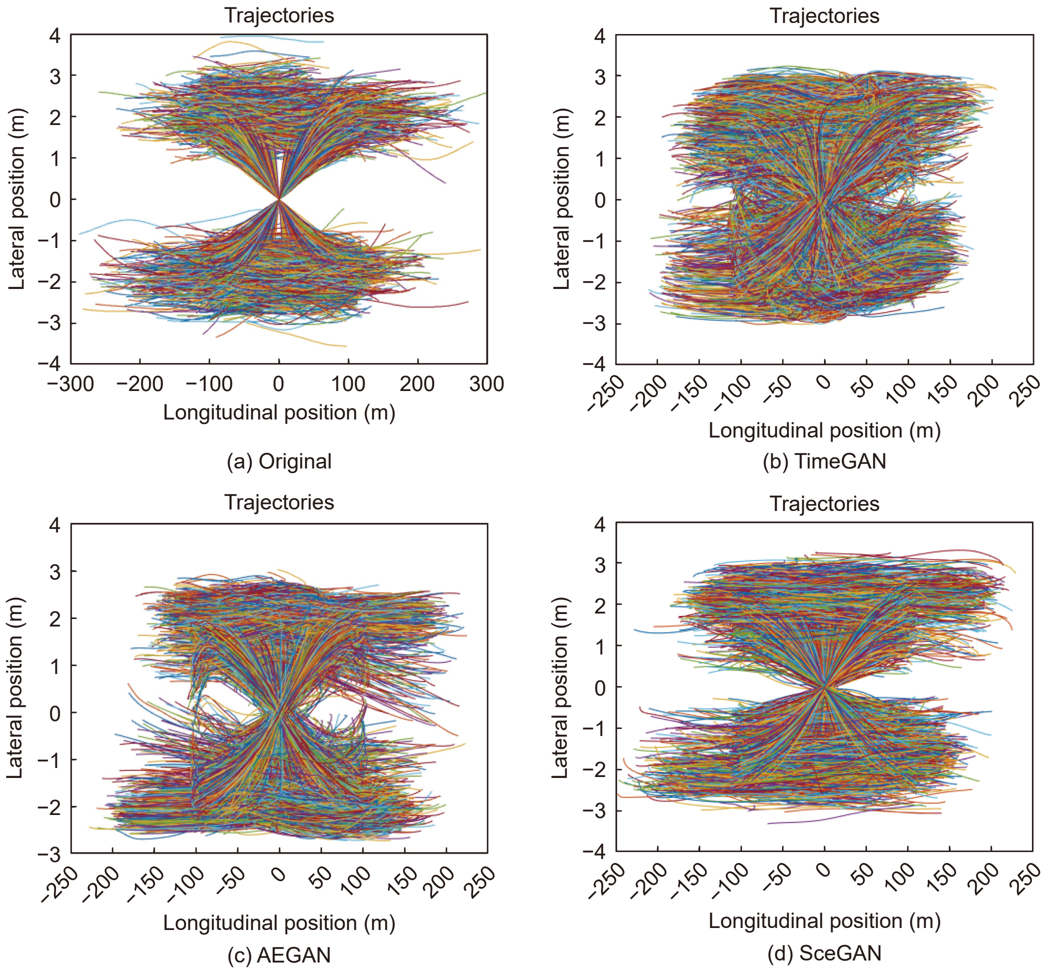


Fig. 5 Trajectories of cut-in scenarios.

Table 4 Numbers and proportion of scenarios generated in different risk levels

Set	High risk		Medium risk		Low risk		Safe scenario	
	Num	Prop	Num	Prop	Num	Prop	Num	Prop
Original	28	1.10%	103	4.04%	1,299	51.00%	1,117	43.86%
2,547	31	1.22%	109	4.28%	1,296	50.88%	1,111	43.62%
3,000	39	1.30%	127	4.23%	1,537	51.23%	1,297	43.23%
5,000	66	1.32%	215	4.30%	2,572	51.44%	2,147	42.94%
10,000	136	1.36%	434	4.34%	5,086	50.86%	4,344	43.44%
20,000	272	1.36%	876	4.38%	10,134	50.67%	8,718	43.59%
50,000	670	1.34%	2,185	4.37%	25,630	51.26%	21,515	43.03%

by 27.22% and 21.39%.

3) The scenarios generated by SceGAN maintained proportions of intervals when the inverse of TTC is used to measure the risk level. Moreover, IOU and t-SNE were combined to quantify coverage, and the results indicated 79.84% coverage for the same number of scenarios generated, which increased to 93.98% with 50,000 scenarios generated.

SceGAN primarily demonstrates its effectiveness in generating cut-in scenarios. This method has the capability to generate a variety of other scenarios, including multivehicle cut-ins, overtaking maneuvers, and car-following scenarios. Achieving this versatility requires adjusting the input attributes and features to suit specific circumstances.

SceGAN achieves a significant increase in the number of scenarios, and the generated data are all over each risk interval, which solves the problems of insufficient numbers and low quality of existing test scenarios. However, one of the difficulties in autonomous vehicle testing is the specific discovery and generation of safety-critical scenarios; subsequent research should focus on designing and implementing conditional generation test scenario methods. In addition, during actual driving, various environmental factors may influence autonomous vehicles, including weather conditions and traffic infrastructure. Consequently, other factors should be considered in the attributes of SceGAN, and a scenario library with different complexities and driving behaviors should be constructed.

Replication and data sharing

The highD dataset is available at https://levelxdata.com/highd-dataset.

https://doi.org/10.26599/JICV.2023.9210023

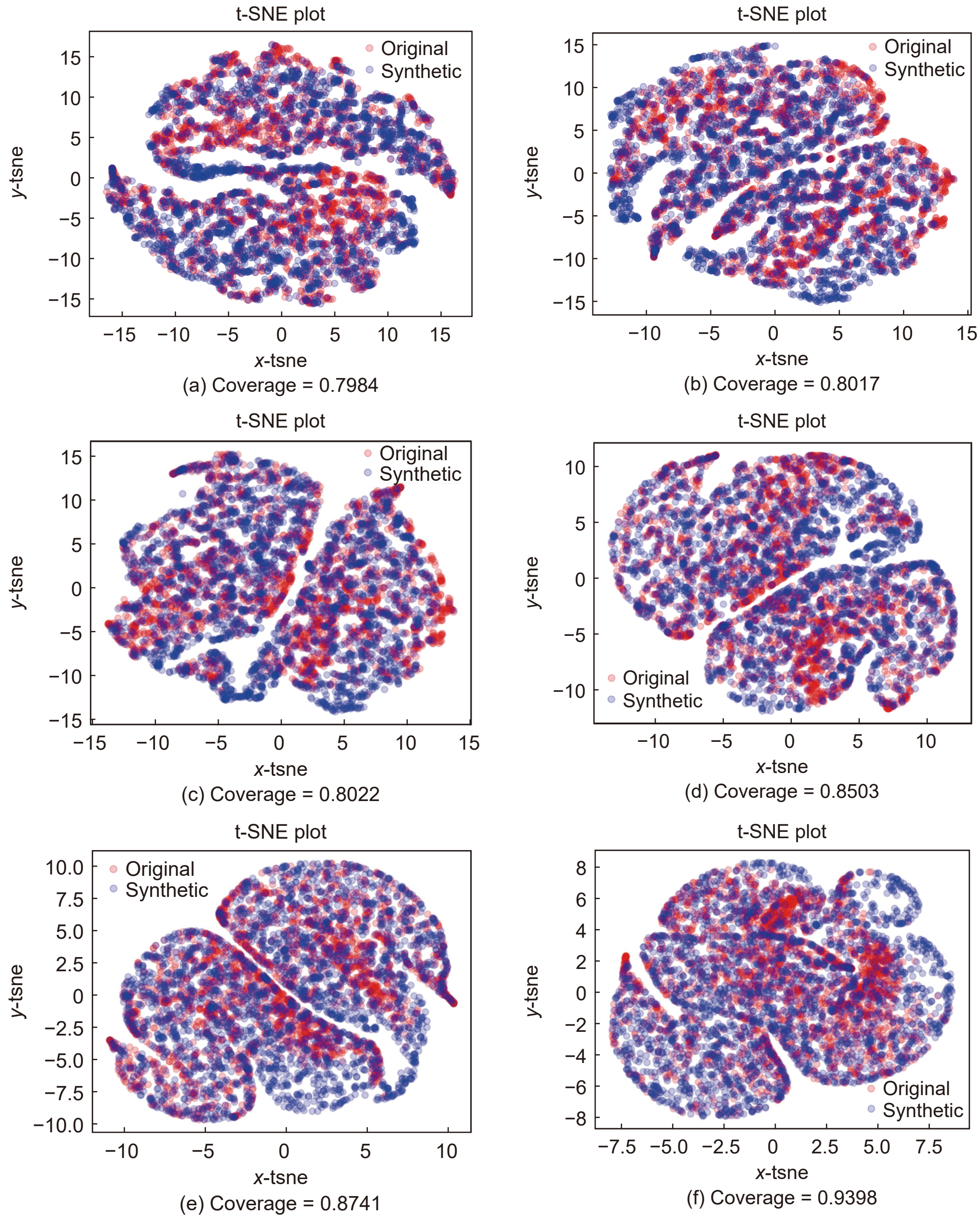


Fig. 6 T-SNE visualization and coverage values, corresponding to the number of generated scenarios: (a) 2,547, (b) 3,000, (c) 5,000, (d) 10,000, (e) 20,000, and (f) 50,000.

Acknowledgements

This research was supported by the National Key R&D Program of China (2021YFB2501200), the National Natural Science Foundation of China (52131204), and the Shaanxi Province Key Research and Development Program (2022GY-300).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

Barz, B., Rodner, E., Garcia, Y. G., Denzler, J., 2019. Detecting regions of maximal divergence for spatio-temporal anomaly detection. IEEE Trans Pattern Anal Mach Intell, 41, 1088−1101.

Demetriou, A., Alfsvåg, H., Rahrovani, S., Haghir Chehreghani, M., 2023. A deep learning framework for generation and analysis of driving scenario trajectories. SN Comput Sci, 4, 1−14.

Duan, J., Gao, F., He, Y., 2022. Test scenario generation and optimization technology for intelligent driving systems. IEEE Intell Transport Syst Mag, 14, 115−127.

Essa, M., Sayed, T., 2019. Full Bayesian conflict-based models for real time safety evaluation of signalized intersections. Accid Anal Prev, 129, 367−381.

Feng, S., Feng, Y., Sun, H., Bao, S., Zhang, Y., Liu, H. X., 2021. Testing scenario library generation for connected and automated vehicles, part II: Case studies. IEEE Trans Intell Transport Syst, 22, 5635−5647.

Feng, S., Sun, H., Yan, X., Zhu, H., Zou, Z., Shen, S., *et al.*, 2023. Dense reinforcement learning for safety validation of autonomous vehicles. Nature, 615, 620−627.

Feng, T., Liu, L., Xing, X., Chen, J., 2022. Multimodal critical-scenarios search method for test of autonomous vehicles. J Intell Connect Veh, 5, 167−176.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A., 2017. Improved training of Wasserstein GANs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 5769–5779.

He, Y., Liu, Y., Yang, L., Qu, X., 2023. Deep adaptive control: Deep reinforcement learning-based adaptive vehicle trajectory control algorithms for different risk levels. IEEE Trans Intell Veh. Http://doi.org/10.1109/TIV.2023.3303408.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput, 9, 1735–1780.

Hoseini, F., Rahrovani, S., Chehreghani, M. H., 2021. Vehicle motion trajectories clustering via embedding transitive relations. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 1314–1321.

Hu, X., Zhu, B., Tan, D., Zhang, N., Wang, Z., 2022. Test scenario generation method for autonomous vehicles based on combinatorial testing and Bayesian network. Proc Inst Mech Eng Part D J Automob Eng, 095440702211255.

Jenkins, I. R., Gee, L. O., Knauss, A., Yin, H., Schroeder, J., 2018. Accident scenario generation with recurrent neural networks. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 3340–3345.

Krajewski, R., Moers, T., Nerger, D., Eckstein, L., 2018a. Data-driven maneuver modeling using generative adversarial networks and variational autoencoders for safety validation of highly automated vehicles. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2383–2390.

Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018b. 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), 2118–2125.

Li, P., Jin, S., Hu, W., Gao, L., Che, Y., Tan, Z., Dong, X., 2022. Complexity evaluation of vehicle-vehicle accident scenarios for autonomous driving simulation tests. J Automo Saf Energy, 13, 697–704. (in Chinese)

Li, Y., Tao, J., Wotawa, F., 2020. Ontology-based test generation for automated and autonomous driving functions. Inf Softw Technol, 117, 106200.

Lin, H., Liu, Y., Li, S., Qu, X., 2023. How generative adversarial networks promote the development of intelligent transportation systems: A survey. IEEE/CAA J Autom Sinica, 10, 1781–1796.

Liu, R. W., Guo, Y., Lu, Y., Chui, K. T., Gupta, B. B., 2022a. Deep network-enabled haze visibility enhancement for visual IoT-driven intelligent transportation systems. IEEE Trans Ind Inf, 19, 1581–1591.

Liu, Y., Wu, F., Lyu, C., Li, S., Ye, J., Qu, X., 2022b. Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform. Transp Res Part E Logist Transp Rev, 161, 102694.

Liu, Y., Wu, F., Liu, Z., Wang, K., Wang, F., Qu, X., 2023. Can language models be used for real-world urban-delivery route optimization? Innovation, 4, 100520.

Maaten, L.V., Hinton, G.E., 2008. Visualizing Data using t-SNE. J Mach Learn Res, 9, 2579–2605.

Pei, H., Ren, K., Yang, Y., Liu, C., Qin, T., Li, D., 2021. Towards generating real-world time series data. In: 2021 IEEE International Conference on Data Mining (ICDM), 469–478.

Rocklage, E., Kraft, H., Karatas, A., Seewig, J., 2017. Automated scenario generation for regression testing of autonomous vehicles. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 476–483.

Shu, H., Yuan, K., Xiu, H. L., Xia, Q., He, S., 2019. Construction of basic test scenarios of automated vehicles. China J Highw Transp, 32, 245–254. (in Chinese)

Tan, S., Wong, K., Wang, S., Manivasagam, S., Ren, M., Urtasun, R., 2021. SceneGen: learning to generate realistic traffic scenes. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 892–901.

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., Maurer, M., 2015. Defining and substantiating the terms scene, situation, and scenario for automated driving. In: 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), 982–988.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., *et al.*, 2017. Attention is all You need. https://arxiv.org/abs/1706.03762.pdf

Xu, Y., Zou, Y., Sun, J., 2018. Accelerated testing for automated vehicles safety evaluation in cut-in scenarios based on importance sampling, genetic algorithm and simulation applications. J Intell Connect Veh, 1, 28–38.

Yang, Y., Wang, Y., Yin, C., Ji, Q., 2021. Simulation testing scenario generation for comfort evaluation of automated vehicles. In: 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI), 1–6.

Yoon, J., Jarrett, D., Van der Schaar, M., 2019. Time-series generative adversarial networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems, 5508–5518.

Yu, J., Jiang, Y., Wang, Z., Cao, Z., Huang, T., 2016. UnitBox: An advanced object detection network. In: Proceedings of the 24th ACM international conference on Multimedia, 516–520.

Zhu, Y., Zhao, X. M., Xu, Z. G., Wang, R. M., 2022. Automatic generation algorithm of lane-change virtual test scenario on highways for automated vehicles using Monte Carlo simulation. China J Highw Transp, 35, 89–100. (in Chinese)

Zhou, W. S., Zhu, Y., Zhao, X. M., Wang, R. M., Xu, Z. G., 2021. Vehicle cut-in test case generation methods for testing of autonomous driving on highway. Automob Technol, 1, 11–18.

Zong, F., Wang, M., Tang, J., Zeng, M., 2022a. Modeling AVs & RVs' car-following behavior by considering impacts of multiple surrounding vehicles and driving characteristics. Phys A Stat Mech Appl, 589, 126625.

Zong, F., He, Z., Zeng, M., Liu, Y., 2022b. Dynamic lane changing trajectory planning for CAV: A multi-agent model with path preplanning. Transp B Transp Dyn, 10, 266–292.

Lan Yang received the Ph.D. degree from Chang'an University, Xi'an, China, in 2013. Currently, she is an Associate Professor with the School of Information Engineering, Chang'an University. Her research interests include the cooperative eco-driving of CAVs, vehicle active safety technology, and vehicle behavior cognition.

Jiaqi Yuan received the B.S. degree in software engineering from School of Computer Science, Chengdu University, Chengdu, China, in 2021. She is currently pursuing the M.Eng. degree in the School of Information Engineering at Chang'an University, Xi'an, China. Her research interests include autonomous driving test and evaluation, deep learning, and intelligent transportation systems.

Xiangmo Zhao (Member, IEEE) received the Ph.D. degree from Chang'an University, Xi'an, China, in 2006. He is a Professor with the School of Information Engineering, Chang'an University. He is also the Vice President of the Joint Laboratory for Connected Vehicles, Ministry of Education and China Mobile, and the Shaanxi Road Traffic Intelligent Detection and Equipment Engineering Technology Research Centre. His research interests include connected vehicles, automated vehicles, and intelligent transportation systems. He is the Director of the Information Professional Committee, a member of the Advisory Expert Group of the China Transportation Association, a member of the National Motor Vehicle Operation Safety Testing Equipment Standardization Committee and the Leading Group of the National Traffic Computer Application Network, the Vice Chairperson of the Institute of Highway Association on Computer Professional Committee, and the Deputy Director of the Institute of Computer in Shaanxi Province.

Shan Fang received the B.S. degree in computer science from Jilin University, Changchun, China, in 2016. He is currently pursuing the Ph.D. degree in transportation information engineering and control in the School of Information Engineering, Chang'an University, Xi'an, China. His research interests include traffic flow simulation and cooperative eco-driving strategy for vehicles.

Zeyu He received the B.E. degree in network engineering from Tianjin University of Finance and Economics, Tianjin, China, in 2021. He is currently pursuing the M.Eng. degree in the School of Information Engineering at Chang'an University, Xi'an, China. His research interests include autonomous vehicle environment perception, object detection, and deep learning.

Jiahao Zhan received the B.E. degree in electronic information engineering from Nanchang Hangkong University, Nanchang, China, in 2021. He is currently pursuing the M.Eng. degree in transportation engineering with Chang'an University, China. His research interests include decision-making and the control of automated vehicle.

Zhiqiang Hu received the B.S. degree in rail transit signal and control from Hubei Engineering University, Xiaogan, China, in 2022. He is currently pursuing the M.Eng. degree in the School of Transportation Engineering at Chang'an University, Xi'an, China. His research interests include autonomous driving decision planning, deep reinforcement learning, and intelligent transportation systems.

Xia Li received the B.S. degree in electronic information science and technology from the School of Physics and Electronic Science, Datong University, in 2022. She is currently pursuing the M.Eng. degree in the School of Information Engineering at Chang'an University, Xi'an, China. Her research interests include autonomous driving test and evaluation, inverse reinforcement learning, and intelligent transportation systems.