Delft University of Technology

Delft Center for Systems and Control

Technical report 22-012

Scenario Parameter Generation Method and Scenario Representativeness Metric for Scenario-Based Assessment of Automated Vehicles*

E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg, and B. De Schutter

To cite this work, please refer to the published version:

E. de Gelder, J. Hof, E. Cator, J.-P. Paardekooper, O. Op den Camp, J. Ploeg, and B. De Schutter, "Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18794–18807, Oct. 2022. doi:10.1109/TITS.2022.3154774

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://dspub.eu/22-012

Scenario Parameter Generation Method and Scenario Representativeness Metric for Scenario-Based Assessment of Automated Vehicles

Erwin de Gelder[1,2*], Jasper Hof[3], Eric Cator[4], Jan-Pieter Paardekooper[1,5], Olaf Op den Camp[1], Jeroen Ploeg[6], Bart De Schutter[2]

***Abstract*—The development of assessment methods for the performance of Automated Vehicles (AVs) is essential to enable the deployment of automated driving technologies, due to the complex operational domain of AVs. One candidate is scenario-based assessment, in which test cases are derived from real-world road traffic scenarios obtained from driving data. Because of the high variety of the possible scenarios, using only observed scenarios for the assessment is not sufficient. Therefore, methods for generating additional scenarios are necessary.**

Our contribution is twofold. First, we propose a method to determine the parameters that describe the scenarios to a sufficient degree while relying less on strong assumptions on the parameters that characterize the scenarios. By estimating the probability density function (pdf) of these parameters, realistic parameter values can be generated. Second, we present the Scenario Representativeness (SR) metric based on the Wasserstein distance, which quantifies to what extent the scenarios with the generated parameter values are representative of real-world scenarios while covering the actual variety found in the real-world scenarios.

A comparison of our proposed method with methods relying on assumptions of the scenario parameterization and pdf estimation shows that the proposed method can automatically determine the optimal scenario parameterization and pdf estimation. Furthermore, it is demonstrated that our SR metric can be used to choose the (number of) parameters that best describe a scenario. The presented method is promising, because the parameterization and pdf estimation can directly be applied to already available importance sampling strategies for accelerating the evaluation of AVs.

I. Introduction

An essential aspect in the development of cro:avAutomated Vehicles (AVs) is the assessment of the quality and performance of AV behavior with respect to safety, comfort, and efficiency [1, 2, 3]. Because public road tests are expensive and time consuming [4, 5], a scenario-based approach has been proposed [2, 6, 7, 8, 9, 10, 11]. With a scenario-based approach, the response of the system-under-test is assessed in many scenarios and for the variations of these scenarios that occur in the real world. Here, a scenario describes the situation the system-under-test is in and how this situation develops over time (in Section III-A, a precise definition of the term scenario is provided). One of the advantages of a scenario-based approach is that the assessment can focus on the more challenging situations by selecting scenarios that are challenging for the system-under-test. As a source of information for the assessment scenarios, real-world driving data has been proposed, thereby guaranteeing that the scenarios represent real-world driving conditions [7, 8, 9].

For the scenario-based assessment approach, it is important that the generated scenarios are representative of scenarios that could happen in real life. In other words, the scenarios should be a representation of the real world [6]. Only then, the results of the assessment can be generalized to the performance of the system-under-test when operating in real life [10]. Furthermore, it is essential that the generated scenarios cover the same variety that is found in real life. Riedmaier et al. [6] argue that since an infinite number of situations occur in the real world, the scenario generation methods must provide a large number of variations in order to cover this infinite number of situations.

Our data-driven approach uses observed scenarios to generate parameter values that describe new scenarios. Instead of relying on a predetermined functional form of the signals, such as a vehicle's speed, and fitting parameters to this functional form, we employ a cro:svdSingular Value Decomposition (SVD) [12] to determine in a data-driven manner the parameters that best describe the scenarios. Next, the cro:pdfprobability density function (pdf) of the parameters is estimated, such that the pdf can be used to sample the parameters to generate similar scenarios. To not assume a particular shape of the pdf, cro:kdeKernel Density Estimation (KDE) [13, 14] is used for the pdf estimation. Furthermore, with KDE, the correlations that might exist among the parameters is modeled. This work also proposes a novel metric called the cro:srScenario Representativeness (SR) metric for quantifying to what extent the generated scenarios are representative and cover the actual variety of real-world scenarios. More specifically, this metric uses the Wasserstein distance [15] to compare a set of generated scenarios with a set of observed

[1]TNO, Integrated Vehicle Safety, Helmond, The Netherlands
[2]Delft University of Technology, Delft Center for Systems and Control, Delft, The Netherlands
[3]Radboud University Medical Center, Department for Health Evidence, Nijmegen, The Netherlands
[4]Radboud University, Applied Stochastics, Nijmegen, The Netherlands
[5]Radboud University, Donders Institute for Brain, Cognition and Behaviour, Nijmegen, The Netherlands
[6]Eindhoven University of Technology, Dept. of Mechanical Engineering Dynamics and Control group, Eindhoven, The Netherlands
*Corresponding author.
E-mail address: erwin.degelder@tno.nl

scenarios.

This article is organized as follows. Section II reviews related works. In Section III, the approach for generating scenarios for the assessment of AVs is explained. Next, Section IV presents a novel metric for quantifying the performance of the scenario-generation method. A case study is performed in Section V. Section VI discusses relevant implications of our approach and some directions for future research. Conclusions of the paper are provided in Section VII.

II. Related works

In this section, first, works concerning the generation of scenarios for the assessment of AVs are reviewed. Next, works related to the SR metric are reviewed.

A. Scenario generation

The approaches to determine scenarios for the assessment of AVs can be categorized into three kinds [16]: scenarios based on observations of real-world traffic, scenarios based on the functionality that is being assessed, and a combination of these two approaches. The current paper focuses on the first approach.

In the literature, several methods are proposed to generate scenarios for the assessment based on real-world driving data. Lages et al. [17] proposed a method to construct scenarios in a virtual simulation environment by reconstructing the real-world scenarios observed by laser scanners. Zofka et al. [18] presented how recorded sensor data can be exploited to create scenarios that might lead to critical situations by modifying parameters of the recorded parameterized scenarios. Stepien et al. [19] generate scenarios by sampling scenario parameter values from generalized extreme value distributions, where the distribution parameters are fitted using scenario parameter values extracted from safety-critical scenarios observed in naturalistic driving data. In [10, 20, 21, 22, 23, 24], also parameterized scenarios were generated and, in addition, importance sampling techniques were presented that automatically generate scenarios in which the system-under-test shows (safety-)critical behavior. Other approaches to generate scenarios in which the system-under-test shows (safety-)critical behavior are Monte Carlo tree search [25] and genetic programming [26]. Schuldt et al. [27] provided a method to generate scenarios using combinatorial algorithms that should ensure that the test cases cover the variety of the possible situations the system-under-test could encounter in real life. More recently, Spooner et al. [28] presented a cro:ganGenerative Adversarial Network (GAN) to generate pedestrian crossing scenarios.

In the existing literature, the scenario generation methods for the assessment of AVs have either one or more of the following shortcomings:

- Observed scenarios are replayed without adding more variations [17]. In this case, the total variety of scenarios that is found in real life will not be covered unless unrealistic amounts of data are gathered.
- The scenarios are oversimplified. For example, a vehicle's speed profile follows a predetermined functional form [10, 19, 21].
- Assumptions regarding the scenario parameter distributions are made that potentially compromise the quality of the scenarios. For example, the parameters are assumed to originate from a Gaussian [29] or generalized extreme value [19] distribution, and/or it is assumed that (some of) the parameters are uncorrelated [24].
- Because no pdf of the scenario parameters is known [18], no evaluation can be made of the performance of the system once deployed on the road, because it is unknown how realistic and likely the scenarios are.

In Section III, a method is proposed that overcomes these shortcomings.

B. Scenario representativeness metric

The generated scenarios should represent scenarios that could happen in real life. Whereas different approaches exist in the literature regarding the generation of scenarios for the assessment of AVs, less is known about the comparison of the generated scenarios with real-life traffic. From the mentioned sources in Section II-A, only Feng et al. [24] compared their generated scenarios with the ground truth from naturalistic driving data. Feng et al. [24] compared the distributions of vehicle speeds and bumper-to-bumper distances between the constructed scenarios and the ground truth. To quantify the similarity between the distributions, the Hellinger distance [30] and the mean absolute error were used. The disadvantages of this approach are that:

1) the generated scenarios may still be substantially different even though the distributions of the vehicle speeds and bumper-to-bumper distances are similar, and
2) only the marginal distributions are considered while the correlation between the vehicle speeds and bumper-to-bumper distances might be completely different.

Whereas little is known about comparing the generated scenarios for the scenario-based assessment of AVs with ground truth data, many similarity metrics for comparing two pdfs are known [30]. Well-known metrics are the Minkowski metric [30], which is a generalized version of the Euclidean distance, the f-divergence, which is a generalized version of both the Kullback-Leibler divergence [31] and the Hellinger distance [30], and the Wasserstein metric [15]. For practical reasons, this work uses the Wasserstein metric. As is shown in Section IV-B, the Wasserstein distance can be estimated using empirical distributions, i.e., without the need to estimate and evaluate a pdf. The other mentioned metrics require integration over the domain of the pdfs, which will give computational issues since the considered pdfs will have a high dimensionality.

III. Scenario generation

To generate realistic scenarios for the assessment of AVs, we use a data-driven approach: observed scenarios are used to generate new scenarios. To do this, the scenarios are parameterized, i.e., parameters are defined that characterize a scenario. For example, the duration of a scenario could be a parameter.

Next, the pdf of the parameters is estimated. This pdf can be used to generate parameter values for new scenarios. In addition, the pdf contains the statistical information of the parameters so that the performance of AVs can be estimated [10, 32]. Choosing the parameters that describe a scenario, however, is not trivial:

- Choosing too few parameters might lead to an oversimplification of the actual scenarios. As a result, not all possible variations of a scenario are modeled.
- Too many parameters lead to problems with estimating the pdf, due to the curse of dimensionality [33].

To overcome this problem, we first consider as many parameters as needed for a complete description of the scenarios to avoid the oversimplification of the scenarios. Next, using an SVD, a new set of parameters is created using a linear mapping of the original scenario parameters. Because this new set of parameters is ordered according to the contribution of each of these parameters in describing the variation that exists among the original scenario parameters, we will consider only the most important parameters without losing too much information. In this way, the curse of dimensionality is avoided without relying on a predetermined choice of parameters.

Below, we first explain how to describe a scenario using many parameters. Next, Section III-B proposes the use of the SVD to reduce the number of parameters. Section III-C describes how KDE is used to estimate the pdf of the reduced set of parameters and how the estimated KDE can be used to generate new scenarios.

A. Parameterization of scenarios

The first step of our approach is the parameterization of scenarios. There is no single best way to parameterize the scenarios considering the wide variety of scenarios. To deal with this variety, this work distinguishes quantitative scenarios from qualitative scenarios, using the definitions of *scenario* and *scenario category* of [34]:

Definition 1 (Scenario)**.** *A scenario is a quantitative description of the relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, the dynamic environment and all events that are relevant to the ego vehicle(s) within the time interval between the first and last relevant event.*

Definition 2 (Scenario category)**.** *A scenario category is a qualitative description of relevant characteristics and activities and/or goals of the ego vehicle(s), the static environment, and the dynamic environment.*

A scenario category is an abstraction of a scenario and, therefore, a scenario category comprises multiple scenarios [34]. For example, the scenario category "cut-in" comprises all possible cut-in scenarios. The goal of our approach is to determine the optimal parameterization of scenarios of a given scenario category based on a set of observed scenarios of the same scenario category and to estimate the pdf of these parameters that can be used to generate parameter values for new scenarios.

The observed scenarios are described using a time series for the content of the scenario that changes within the time window of the scenario (e.g., the speed of a vehicle) and some additional parameters for the content that is fixed (e.g., the lane width and the duration of the scenario). Here, $y(t) \in \mathbb{R}^{n_y}$ denotes the time series of a scenario with $t \in [t_0, t_1]$, where n_y denotes the dimension of the time series and t_0 and t_1 denote the start and end time of the scenario, respectively. The n_θ additional parameters are represented by $\theta \in \mathbb{R}^{n_\theta}$.

To deal with the time series, the continuous time interval $[t_0, t_1]$ is discretized, such that two consecutive time instants are $(t_1 - t_0)/(n_\mathrm{t} - 1)$ apart. This gives:

$$\mathbf{y} = \begin{bmatrix} y(t_0) \\ y\Big(t_0 + \frac{t_1 - t_0}{n_\mathrm{t} - 1}\Big) \\ y\Big(t_0 + 2\frac{t_1 - t_0}{n_\mathrm{t} - 1}\Big) \\ \vdots \\ y(t_1) \end{bmatrix} \in \mathbb{R}^{n_\mathrm{t} n_y}. \tag{1}$$

Note that n_t must be chosen such that no important information is lost during the discretization. Because in practice, due to the discrete nature of sensor readings, the time series $y(t)$ is obtained at certain specific times rather than on a continuous time interval, it may be required to use interpolation techniques, such as splines [35], to evaluate $\mathbf{y}$.

Let us assume that N_x observed scenarios can be used to generate new scenarios. To indicate that the scenario parameters $\mathbf{y}$ and θ belong to a specific scenario, the index $i \in \{1, \ldots, N_x\}$ is used, i.e., the parameters of the i-th scenario are $\mathbf{y}_i$ and θ_i. To further ease the notation, $\mathbf{y}_i$ and θ_i are combined into one vector $\mathbf{x}_i$:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{y}_i \\ \theta_i \end{bmatrix} \in \mathbb{R}^{n_\mathrm{t} n_y + n_\theta}. \tag{2}$$

B. Parameter reduction using Singular Value Decomposition

As shown in (2), $n_x = n_\mathrm{t} n_y + n_\theta$ parameters describe a scenario. Even for small numbers of n_t, n_y, and n_θ, the total number of parameters becomes too large to reliably estimate the joint pdf. One way to avoid this curse of dimensionality is to assume that the parameters are independent, but especially the parameters $y(t_0)$ till $y(t_1)$ in (1) are obviously correlated, so assuming that the parameters are independent is not a good solution.

In the field of machine learning, cro:pcaPrincipal Component Analysis (PCA) is commonly used for dimensionality reduction [36]. As PCA uses the SVD [12], this work uses the SVD to transform the parameters $\mathbf{x}_i$ into a lower-dimensional vector of parameters. Before applying the SVD, the parameters are weighted with $\alpha \in \mathbb{R}^{n_x}$ in order to give more or less importance to the n_x parameters. This is particularly useful to compensate for the imbalance in the parameter vector, where the imbalance is caused by the fact that the parameter vector considers the time series $y(t)$ at n_t different times and the additional parameters θ only once. Let us define a matrix that contains the parameters of the N_x scenarios:

$$X = \begin{bmatrix} (\alpha \odot \mathbf{x}_1) - \mu & \cdots & (\alpha \odot \mathbf{x}_{N_x}) - \mu \end{bmatrix} \in \mathbb{R}^{n_x \times N_x}, \tag{3}$$

where $\odot$ denotes the element-wise product of vectors and $\mu \in \mathbb{R}^{n_x}$ denotes the mean of the weighted scenario parameters:

$$\mu = \frac{1}{N_x} \sum_{i=1}^{N_x} \alpha \odot \mathbf{x}_i. \tag{4}$$

Using the SVD of X, we obtain:

$$X = U \Sigma V^{\mathsf{T}}. \tag{5}$$

Here, both $U \in \mathbb{R}^{n_x \times n_x}$ and $V \in \mathbb{R}^{N_x \times N_x}$ are orthonormal matrices. Therefore, both matrices can be interpreted as rotation matrices in $\mathbb{R}^{n_x}$ and $\mathbb{R}^{N_x}$, respectively. The matrix $\Sigma \in \mathbb{R}^{n_x \times N_x}$ takes the same shape as X. This matrix has only zeros except on (part of) the diagonal. The diagonal contains the so-called singular values, denoted by σ_j with $j \in \{1, \ldots, \bar{N}\}$, $\bar{N} = \min(n_x, N_x)$. These singular values are in decreasing order, i.e.,

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\bar{N}} \geq 0. \tag{6}$$

Because of the decreasing singular values, rotating the matrix X from the left with U^{T} transforms the data to a new coordinate system such that the first coordinate has the largest variance compared to the other coordinates. This variance equals σ_1^2. Similarly, the second largest variance equals σ_2^2 and lies on the second coordinate, etc. Because of the decreasing variance, the scenario parameters can be approximated using only the first d coordinates of the new coordinate system, as these d coordinates describe the majority of the variations. So, the scenario parameters of the i-th scenario are approximated by setting $\sigma_j = 0$ for $j > d$:

$$\alpha \odot \mathbf{x}_i = \mu + \sum_{j=1}^{\bar{N}} \sigma_j v_{ij} \mathbf{u}_j \approx \mu + \sum_{j=1}^{d} \sigma_j v_{ij} \mathbf{u}_j, \tag{7}$$

where v_{ij} is the (i,j)-th element of V, $\mathbf{u}_j$ is the j-th column of U, and d is the number of parameters that are retained. Thus, the n_x parameters of the i-th scenario are approximated using the d parameters $v_{i1}, \ldots, v_{id}$. The singular values $\sigma_1, \ldots, \sigma_d$, the vectors $\mathbf{u}_1, \ldots, \mathbf{u}_d$, and μ are used to map the new scenario parameters, $v_{i1}, \ldots, v_{id}$, to an approximation of the weighted original scenario parameters, $\alpha \odot \mathbf{x}_i$.

Remark 1. Using the approximation of (7), it is not necessary to evaluate the complete SVD of (5). Only the first d columns of U and V need to be computed and only the first d singular values. In practice, $d \ll \bar{N}$, so this saves a significant amount of computation time. $\lozenge$

The choice of $d < \bar{N}$ is not trivial. Choosing d too small results in too much loss of detail. Choosing d too large will give problems when estimating the pdf of the new parameters. One method to choose d is to look at the amount of overall variance of $\alpha \odot \mathbf{x}_i$ explained by the first d singular values. The overall variance scales with the sum of the squared singular values [12, p. 77], i.e.,

$$\sum_{i=1}^{N_x} ((\alpha \odot \mathbf{x}_i) - \mu)^{\mathsf{T}} ((\alpha \odot \mathbf{x}_i) - \mu) = \sum_{j=1}^{\bar{N}} \sigma_j^2. \tag{8}$$

Thus, the first d singular values explain

$$\frac{\sum_{j=1}^{d} \sigma_j^2}{\sum_{j=1}^{\bar{N}} \sigma_j^2} \tag{9}$$

of the overall variance. One approach would be to set d such that (9) exceeds a certain threshold, such as 0.95. Another way to choose d is by inspecting the actual approximation error in (7) and keep increasing d until the approximation error is not too large. Section IV proposes an alternative way to determine d using a metric that quantifies the goal of our generated scenarios, i.e., that the generated scenarios are representing real-world scenarios and cover the actual variety of real-world scenarios.

C. Estimating the probability density function

Using the approximation of (7) based on the SVD, the i-th scenario is described by the vector $\tilde{\mathbf{v}}_i$:

$$\tilde{\mathbf{v}}_i^{\mathsf{T}} = \begin{bmatrix} v_{i1} & \cdots & v_{id} \end{bmatrix}. \tag{10}$$

Note that the d entries of $\tilde{\mathbf{v}}_i$ are linearly uncorrelated with the d entries of $\tilde{\mathbf{v}}_m$ $(m \neq i)$[1]. Despite the linear independence, the different entries of $\tilde{\mathbf{v}}_i$ may still be dependent due to higher-order correlations; so we treat these d entries as dependent variables.

To estimate the pdf of $\tilde{\mathbf{v}}_i$, we propose to use KDE. KDE [13, 14] is often referred to as a non-parametric way to estimate the pdf, because KDE does not rely on the assumption that the data are drawn from a given parametric family of probability distributions. Because KDE produces a pdf that adapts itself to the data, it is flexible regarding the shape of the actual underlying distribution of $\tilde{\mathbf{v}}_i$. In KDE, the pdf is estimated as:

$$\hat{f}_H(v) = \frac{1}{N_x} \sum_{i=1}^{N_x} K_H(v - \tilde{\mathbf{v}}_i). \tag{11}$$

Here, $K_H(\cdot)$ is the so-called scaled kernel with a positive definite symmetric bandwidth matrix $H \in \mathbb{R}^{d \times d}$. The kernel $K(\cdot)$ and the scaled kernel $K_H(\cdot)$ are related using

$$K_H(u) = |H|^{-1/2} K\Big(H^{-1/2} u\Big), \tag{12}$$

where $|\cdot|$ denotes the matrix determinant. The choice of the kernel function is not as important as the choice of the bandwidth matrix [37, 38]. This article considers the Gaussian kernel[2], which is given by

$$K(u) = \frac{1}{(2\pi)^{d/2}} \exp\left\{ -\frac{1}{2} \|u\|_2^2 \right\}, \tag{13}$$

[1]This is assuming that $\sigma_d > 0$. With this assumption and because X in (3) is defined such that the sum of each row of X equals zero, it is easy to verify that $\frac{1}{N_x} \sum_{m=1}^{N_x} v_{mj} = 0$ for $j \in \{1, \ldots, N_x\}$. Therefore, $\sum_{i=1}^{N_x} \Big(v_{ij} - \frac{1}{N_x} \sum_{m=1}^{N_x} v_{mj}\Big)\Big(v_{ik} - \frac{1}{N_x} \sum_{m=1}^{N_x} v_{mk}\Big) = \sum_{i=1}^{N_x} v_{ij} v_{ik} = 0$ for $j \neq k$, where the latter equality follows from the orthonormality of V.

[2]The advantage of the Gaussian kernel is that it gives the possibility to calculate a metric that quantifies the completeness of the data [39] and to apply conditional sampling when generating scenario parameters [40]. Both these topics are out of scope of this article.

where $\|u\|_2^2 = u^\mathsf{T} u$ denotes the squared 2-norm of u.

A bandwidth matrix of the form $H = h^2 I_d$ is used, where I_d denotes the d-by-d identity matrix. The bandwidth h is determined with leave-one-out cross-validation [41], because this minimizes the difference between the real pdf and the estimated pdf according to the Kullback-Leibler divergence [37, 42].

To sample scenario parameters using $\hat{f}_H(\cdot)$, first, an integer $i \in \{1, \ldots, N_x\}$ is randomly chosen with each integer having equal likelihood. Next, a random sample is drawn from a Gaussian with covariance H and mean $\tilde{\mathbf{v}}_i$. Then, using the approximation in (7), the scenario parameters are calculated.

As far as the computational effort is concerned, sampling the scenario parameters from a KDE is efficient because there is no need to actually evaluate the pdf. Determining the optimal bandwidth matrix requires more computational effort, but this only has to be done once per data set. The computational complexity of cross-validation methods for the bandwidth estimation typically scales with N_x^2 [43].

IV. Scenario Representativeness metric

Ideally, the parameters of the generated scenarios are sampled from the same distribution that underlies the real-world scenario parameters. The problem is that this distribution is unknown. Nevertheless, it is possible to define a metric that quantifies the similarity of the distribution that is used to generate scenario parameters and the distribution that underlies the real-world scenario parameters. Section IV-A further explains the goal of this metric, which we call the SR metric. Next, Section IV-B explains the Wasserstein distance [15], which is then applied to derive our metric in Section IV-C.

A. Scenario comparison problem

The set of observed scenarios, described using the parameters $\mathbf{x}_i$, $i \in \{1, \ldots, N_x\}$, are used for generating the scenario parameters. To ease the notation, let us denote the set of observed scenarios by $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N_x}\}$. This work assumes that these scenarios — that are comprised by the same scenario category — are independently and identically distributed according to the distribution $f(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$. Let us denote the set of generated scenario parameter vectors by $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{N_w}\}$ where $\mathbf{w}_i \in \mathbb{R}^{n_x}$, $i \in \{1, \ldots, N_w\}$ are similarly parameterized as in (2) and N_w is the number of generated scenario parameter vectors. Let $\hat{f}(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$ denote the pdf of the generated scenario parameter vectors, which is obtained from $\hat{f}_H(\cdot) : \mathbb{R}^d \to \mathbb{R}$ under a change of variable according to the approximation in (7). As later appears, it is not needed to have an explicit definition for $\hat{f}(\cdot)$. Ideally, $\hat{f}(\cdot)$ is equal to $f(\cdot)$. So our metric aims to quantify the similarity of $\hat{f}(\cdot)$ and $f(\cdot)$.

To estimate the similarity between $\hat{f}(\cdot)$ and $f(\cdot)$, we cannot simply compare $\mathcal{W}$ with $\mathcal{X}$. In that case, taking $\mathcal{W} = \mathcal{X}$ would give us the best result, but this is undesirable because, ideally, the scenarios of the generated parameters cover the whole variety of real-world scenarios and not just the variety that have been observed in $\mathcal{X}$. Therefore, another set of scenarios is needed that can be used to test. Let us assume that such a set of scenarios is available, denoted by $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_{N_z}\}$ where $\mathbf{z}_i \in \mathbb{R}^{n_x}$, $i \in \{1, \ldots, N_z\}$ are independently and identically distributed according to $f(\cdot)$. Thus, $\mathcal{X}$ and $\mathcal{Z}$ can be regarded as a training and test set, respectively.

In summary, the goal is to find a metric that quantifies the similarity of $\hat{f}(\cdot)$ and $f(\cdot)$ using the sets of observed scenario parameters $\mathcal{X}$ and $\mathcal{Z}$ and the set of scenario parameters $\mathcal{W}$, generated based on $\mathcal{X}$.

B. Empirical Wasserstein metric

The p-th Wasserstein metric ($p \geq 1$) [15] is used to compare two pdfs $\xi(\cdot)$ and $\eta(\cdot)$ defined on the set $\mathcal{U}$. This metric is defined as follows:

$$W_p(\xi, \eta) = \left(\inf_{\gamma \in \Gamma(\xi, \eta)} \left\{ \int_{\mathcal{U} \times \mathcal{U}} (\Delta(u, v))^p \, \mathrm{d}\gamma(u, v) \right\} \right)^{1/p}. \tag{14}$$

Here, $\Delta(u, v)$ denotes the distance from u to v, which will be defined below, and $\Gamma(\xi, \eta)$ denotes the set of joint distributions of (u, v) that have marginal distributions $\xi(\cdot)$ and $\eta(\cdot)$. Intuitively, if the pdfs $\xi(\cdot)$ and $\eta(\cdot)$ are seen as two piles of earth having a different shape with mass 1, then (14) calculates the minimum cost of converting one pile of earth with shape $\xi(\cdot)$ into a pile of earth with shape $\eta(\cdot)$. Therefore, the Wasserstein metric is also referred to as the earth mover's distance [44].

In our case, the goal is to have a metric to compare $f(\cdot)$ and $\hat{f}(\cdot)$. Because $f(\cdot)$ is unknown, its approximation based on $\mathcal{Z}$ is considered:

$$f(\mathbf{z}) \approx \frac{1}{N_z} \sum_{i=1}^{N_z} \delta(\mathbf{z} - \mathbf{z}_i), \mathbf{z} \in \mathbb{R}^{n_x}, \tag{15}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Considering the high dimension of $\mathbf{z}$, numerical approximation of the integral of the Wasserstein metric (14) using this approximation and $\hat{f}(\cdot)$ would require so many evaluations of $\hat{f}(\cdot)$ that it becomes computationally infeasible. Therefore, the empirical estimation of the Wasserstein metric (14) is considered, which makes use of the empirical estimation of $\hat{f}(\cdot)$:

$$\hat{f}(\mathbf{w}) \approx \frac{1}{N_w} \sum_{i=1}^{N_w} \delta(\mathbf{w} - \mathbf{w}_i), \mathbf{w} \in \mathbb{R}^{n_x}. \tag{16}$$

Substituting the empirical estimations of (15) and (16) for $\xi(\cdot)$ and $\eta(\cdot)$, respectively, into (14), leads to the so-called empirical Wasserstein metric [45], which is defined as:

$$\tilde{W}_p(\mathcal{Z}, \mathcal{W}) = \left(\inf_{T} \sum_{i=1}^{N_z} \sum_{j=1}^{N_w} (\Delta(\mathbf{z}_i, \mathbf{w}_j))^p T_{ij} \right)^{1/p}, \tag{17}$$

where T_{ij} is the (i,j)-th element of the transportation matrix T that is subject to the following conditions:

$$\sum_{i=1}^{N_z} T_{ij} = \frac{1}{N_w} \quad \forall\, j \in \{1,\ldots,N_w\}, \tag{18}$$

$$\sum_{j=1}^{N_w} T_{ij} = \frac{1}{N_z} \quad \forall\, i \in \{1,\ldots,N_z\}, \tag{19}$$

$$T_{ij} \geq 0 \qquad \forall\, i \in \{1,\ldots,N_z\}, j \in \{1,\ldots,N_w\} \tag{20}$$

For the distance function, we will use the 2-norm of the difference of the scenario parameters after scaling the scenario parameters according to the weights α that we also used in Section III-B:

$$\Delta(\mathbf{z},\mathbf{w}) = \|(\alpha \odot \mathbf{z}) - (\alpha \odot \mathbf{w})\|_2. \tag{21}$$

C. Metric for testing scenario representativeness

The empirical Wasserstein metric $\tilde{W}_p(\mathcal{Z},\mathcal{W})$ is an approximation of the Wasserstein metric $W_p\left(f,\hat{f}\right)$. As one might expect, using an infinite number of scenario parameters, i.e., for $N_z \to \infty$ and $N_w \to \infty$, the empirical Wasserstein metric approaches the Wasserstein metric with probability 1 [45]. The problem is that N_z and N_w are not infinite. In addition, whereas a fairly large number for N_w can be chosen, as it is only limited by the available computational resources, to increase N_z, more data are needed and this is generally expensive. Therefore, this work proposes a metric that is different from (17).

Our proposed SR metric is based on the following intuition: Suppose that $\hat{f}$ is indeed an approximation of f. Because $\mathcal{X}$ and $\mathcal{Z}$ are based on the same underlying pdf, i.e., f, it is expected that $\tilde{W}_p(\mathcal{X},\mathcal{W})$ is similar to $\tilde{W}_p(\mathcal{Z},\mathcal{W})$. If, however, $\tilde{W}_p(\mathcal{X},\mathcal{W})$ is significantly smaller than $\tilde{W}_p(\mathcal{Z},\mathcal{W})$, it suggests overfitting of the training data because the generated scenario parameters are too much skewed towards the training data $\mathcal{X}$. To penalize overfitting of the training data, our SR metric includes a penalty in case $\tilde{W}_p(\mathcal{Z},\mathcal{W})$ is larger than $\tilde{W}_p(\mathcal{X},\mathcal{W})$. Thus, the SR metric becomes:

$$\begin{aligned} M_p(\mathcal{W},\mathcal{Z},\mathcal{X}) = \tilde{W}_p(\mathcal{Z},\mathcal{W}) + \\ \beta\left(\tilde{W}_p(\mathcal{Z},\mathcal{W}) - \tilde{W}_p(\mathcal{X},\mathcal{W})\right). \end{aligned} \tag{22}$$

Here, β is the weight of the penalty. The case study in Section V demonstrates empirically that $M_p(\mathcal{W},\mathcal{Z},\mathcal{X})$ of Section IV-C better correlates with the Wasserstein metric of (14) than the empirical Wasserstein metric of (17) and a method to choose β.

V. Case Study

To illustrate the proposed method for generating the scenario parameters (Section III) and the SR metric (Section IV), these are applied in a case study. Section V-A explains the scenario categories that are considered in the case study and describes the choices that are made regarding the scenario parameterization. Section V-B illustrates the approximation of the original parameterization using the SVD. Next, the scenario parameter generation method is demonstrated in Section V-C. Section V-C also shows that the SR metric Section IV-C can be used to choose d. Our method for generating scenario parameters is compared with other methods in Section V-D. Section V-E demonstrates that the SR metric Section IV-C better correlates with the Wasserstein metric (14) than the empirical Wasserstein metric (17).

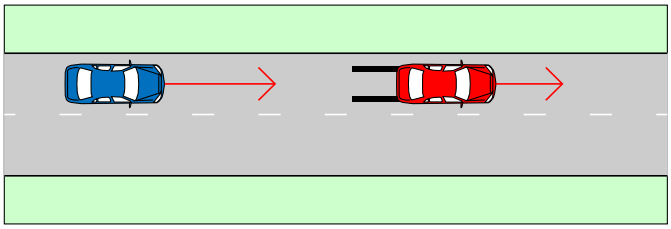

Fig. 1. Schematic representation of the scenario category "leading vehicle decelerating (LVD)". The left vehicle is the ego vehicle.

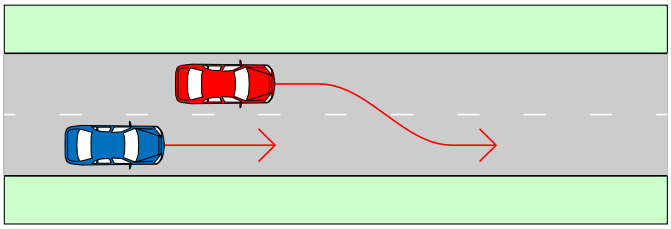

Fig. 2. Schematic representation of the scenario category "cut-in". The left vehicle is the ego vehicle.

A. Scenario categories and parameterization

In this case study, two scenario categories are considered. The first scenario category, labeled cro:lvdleading vehicle decelerating (LVD), involves an ego vehicle that is following another vehicle that decelerates, see Fig. 1. As a result, the ego vehicle might need to brake or change direction to avoid contact with the vehicle that decelerates. The second scenario category considers a vehicle that performs a cut-in, such that this vehicle becomes the leading vehicle of the ego vehicle, see Fig. 2. Depending on the speed and timing of the vehicle that performs a cut-in, the ego vehicle might need to brake or change direction to avoid a collision.

To obtain the scenarios, the data set described in [46] is used. The data were recorded from a single vehicle in which 20 drivers were asked to drive a prescribed route, resulting in 63 hours of data containing 1150 LVD scenarios and 289 cut-in scenarios. The majority of the route was on the highway. To measure the surrounding traffic, the vehicle was equipped with three radars and one camera. The surrounding traffic was measured by fusing the data of the radars and the camera as described in [47]. To extract the LVD and cut-in scenarios from the data set with the fused data, we searched for particular (combinations of) activities in the data: a deceleration activity of a leading vehicle indicates an LVD scenario and a lane

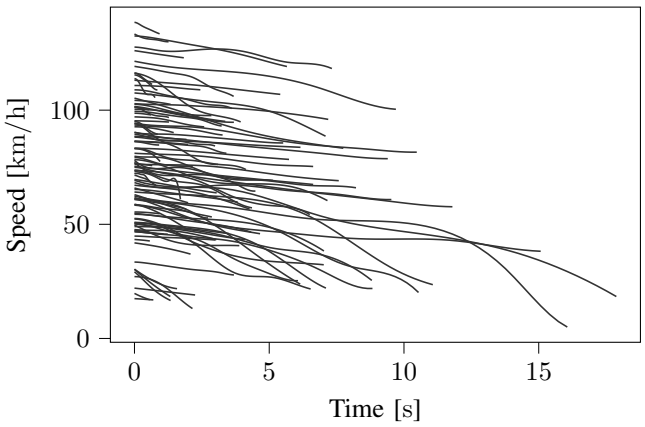

Fig. 3. Speed of the leading vehicle during 100 randomly-selected observed LVD scenarios. For plotting purposes, the starting time of each scenario is set to 0.

change of another vehicle that becomes the leading vehicle indicates a cut-in scenario. For more information on the process of extracting the scenarios, see [48].

From the 1150 LVD scenarios, the training uses $80\,\%$ (so $N_x = 920$) and the testing uses the remaining $20\,\%$ (so $N_z = 230$) as this 80/20 ratio is commonly used for splitting the data into a training set and a test set. The training data are used for generating $N_w = 10000$ new scenario parameter vectors. To describe the decelerating behavior of the leading vehicle, the acceleration of the leading vehicle at $n_\mathrm{t} = 50$ time instants is used ($n_y = 1$). As additional parameters, the duration of the scenario, $t_1 - t_0$, the initial speed of the leading vehicle, and the initial time gap between the leading vehicle and the ego vehicle are considered ($n_\theta = 3$). Thus, $n_x = 53$. In Fig. 3, the speed of the leading vehicle of 100 randomly-selected observed LVD scenarios are shown. The k-th weight, α_k, is obtained by dividing a chosen constant β_k by the standard deviation of the k-th parameter:

$$\alpha_k = \frac{\beta_k}{\sqrt{\frac{1}{N_x}\sum_{i=1}^{N_x}\left(\left(\mathbf{x}_i\right)_k - \bar{\mathbf{x}}_k\right)^2}}, \tag{23}$$

with $\left(\mathbf{x}_i\right)_k$ denoting the k-th element of $\mathbf{x}_i$ and $\bar{\mathbf{x}}_k = \frac{1}{N_x}\sum_{i=1}^{N_x}\left(\mathbf{x}_i\right)_k$. In this way, the contribution of the k-th parameter to the overall variance (see (8)) only depends on β_k. When choosing $\beta_1 = \ldots = \beta_{53}$, the acceleration of the leading vehicle would contribute 50 times more to the overall variance of (8), because $n_\mathrm{t} = 50$ elements are used to describe the acceleration. For the LVD scenarios, we want to give the acceleration the same importance as each of the other parameters, so we choose $\beta_1 = \ldots = \beta_{50} = 1/\sqrt{n_\mathrm{t}}$ and $\beta_{51} = \beta_{52} = \beta_{53} = 1$.

From the 289 cut-in scenarios, $80\,\%$ are used for training (so $N_x = 231$) and $20\,\%$ are used for testing (so $N_z = 58$). Both cut-in scenarios from the left and from the right are considered. The training data are used for generating $N_w = 10000$ parameter vectors that describe cut-in scenarios. A cut-in scenario is described using the speed of the vehicle that performs the lane change and its lateral position with respect to the center of the ego vehicle's lane (so $n_y = 2$) at $n_\mathrm{t} = 50$ time instants. In case of a cut-in scenario from the left, the lateral position is positive when the cutting-in vehicle is on the left of the center of ego vehicle's lane and vice versa for a cut-in scenario from the right. Furthermore, $n_\theta = 3$ extra parameters are used to describe a cut-in scenario: the duration of the scenario, the initial speed of the ego vehicle, and the initial longitudinal position of the cutting-in vehicle with respect to the ego vehicle. Thus, $n_x = 103$. To give the same importance to the speed of the vehicle that performs the lane change, its lateral position, and the 3 extra parameters, the weights are calculated using (23) with $\beta_1 = \ldots = \beta_{100} = 1/\sqrt{n_\mathrm{t}}$ and $\beta_{101} = \beta_{102} = \beta_{103} = 1$.

TABLE I
THE LAST $n_\theta = 3$ COORDINATES OF BOTH μ AND THE FIRST FOUR COLUMNS OF U AFTER SCALING WITH α FOR THE LVD SCENARIOS. NOTE THAT $\oslash$ DENOTES ELEMENT-WISE DIVISION.

	Coordinate 51 Scenario duration	Coordinate 52 Initial speed	Coordinate 53 Initial time gap
$\mu \oslash \alpha$	$4.73\,\mathrm{s}$	$22.11\,\mathrm{km/h}$	$1.49\,\mathrm{s}$
$\mathbf{u}_1 \oslash \alpha$	$-1.50\,\mathrm{s}$	$-15.17\,\mathrm{km/h}$	$0.28\,\mathrm{s}$
$\mathbf{u}_2 \oslash \alpha$	$-3.09\,\mathrm{s}$	$12.22\,\mathrm{km/h}$	$-0.06\,\mathrm{s}$
$\mathbf{u}_3 \oslash \alpha$	$1.15\,\mathrm{s}$	$-16.52\,\mathrm{km/h}$	$0.29\,\mathrm{s}$
$\mathbf{u}_4 \oslash \alpha$	$1.32\,\mathrm{s}$	$-2.88\,\mathrm{km/h}$	$-0.08\,\mathrm{s}$

B. Approximation of scenarios with SVD

As explained in Section III-C, using too many parameters will lead to poor estimations of the pdf of the parameters. We use an SVD to obtain a reduced number of parameters that best describe the original scenarios parameters. This section illustrates the approximation of the original scenario parameters using the parameters obtained after applying the SVD.

Following the approximation of (7), the scaled parameter vector, $\alpha \odot \mathbf{x}_i$, is approximated using a linear combination of the first d columns of U, i.e., $\mathbf{u}_1, \ldots, \mathbf{u}_d$. In Fig. 4 and Table I, μ and the first four columns of U are shown for the LVD scenarios. For an easier interpretation, the original scaling of the parameters by α is undone via the element-wise division by α. Figure 4 shows that the average scenario starts with a deceleration of about $0.4\,\mathrm{m/s^2}$ and ends with a deceleration of about $0.8\,\mathrm{m/s^2}$. Table I shows that the average scenario duration is $4.73\,\mathrm{s}$, the average initial speed of the leading vehicle is $22.11\,\mathrm{km/h}$, and the average initial time gap is $1.49\,\mathrm{s}$. Since each scenario is estimated by combining the curves in Fig. 4 and values in Table I, it can be seen that the approximations do not contain complex acceleration curves. In other words, the accelerations will be smoothed and the details may get lost. The amount of smoothing depends on d, i.e., the number of vectors of U that are used to approximate the original parameter vector. Choosing the value of d is a trade-off: a higher value of d leads to less smoothing and, therefore, a smaller approximation error, but choosing d too large leads to problems when estimating the pdf of the new parameters.

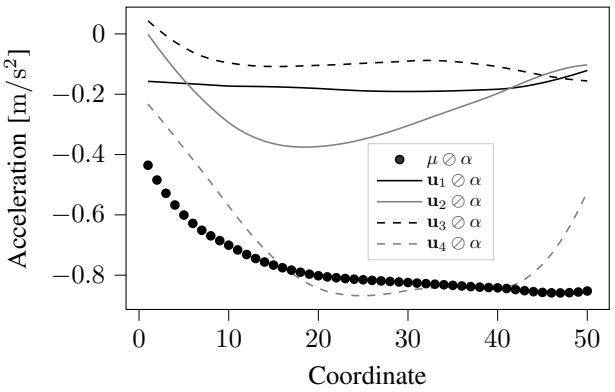

Fig. 4. The first $n_{\mathrm{t}} = 50$ coordinates of both μ and the first four columns of U after scaling with α for the LVD scenarios. Note that $\oslash$ denotes element-wise division.

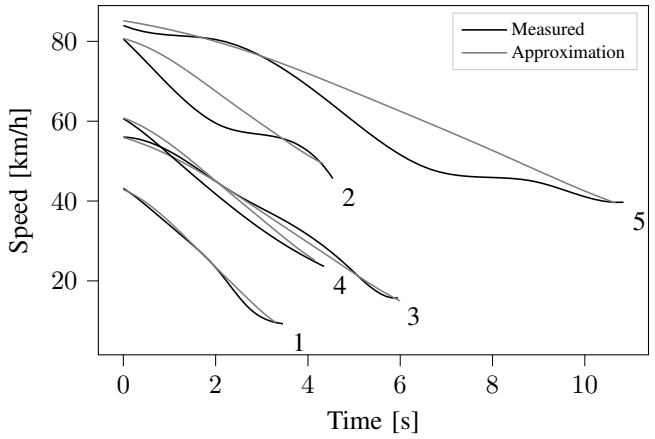

Fig. 5. Five scenarios that require the highest average deceleration of the follower. The black lines denote the observed scenarios and the gray lines denote their approximations based on the $d = 4$ parameters. The corresponding initial time gaps are listed in Table II.

Figure 5 shows five LVD scenarios. These selected LVD scenarios correspond to the five LVD scenarios that require the highest average deceleration of the following vehicle. The line with the "1" denotes the LVD scenario that requires the highest average deceleration. Table II lists the values of $\sigma_j v_{ji}$ for $j \in \{1, \ldots, d\}$ with $d = 4$ that are used to approximate the original scenarios according to the approximation in (7). The gray lines in Fig. 5 show the approximated speed of the five LVD scenarios. Table II shows the initial time gaps of the five scenarios shown in Fig. 5. These five scenarios illustrate that the accelerations are smoothed, but the main characteristics of the scenarios are captured by the approximations: the average deceleration, the scenario duration, the initial speed, and the initial time gap are well approximated.

C. Generating scenario parameters

An important parameter for the generation of the scenario parameter vectors is the number of reduced parameters (d). One approach is to look at the so-called explained variance of (9) of the first d singular values, see Table III. The first 4 singular values already explain $90.4\,\%$ of the variance for the LVD scenarios, so $d = 4$ might be a suitable choice. In Fig. 6, the speed of the leading vehicle of 100 generated LVD scenarios is shown using $d = 4$.

TABLE II
INITIAL TIME GAPS OF THE FIVE SCENARIOS THAT REQUIRE THE HIGHEST AVERAGE DECELERATION OF THE FOLLOWER. THE CORRESPONDING SPEEDS ARE SHOWN IN FIG. 5.

#	$\sigma_1 v_{i1}$	$\sigma_2 v_{i2}$	$\sigma_3 v_{i3}$	$\sigma_4 v_{i4}$	Initial time gap	
					Original	Approximated
1	2.21	0.30	-0.03	2.16	1.91 s	1.91 s
2	-0.28	0.75	-0.46	1.56	1.08 s	1.11 s
3	0.61	-0.21	-0.42	1.53	1.43 s	1.43 s
4	1.74	0.25	0.58	1.63	2.00 s	2.00 s
5	-1.74	-0.71	-0.49	1.30	0.81 s	0.80 s

TABLE III
EXPLAINED VARIANCE ACCORDING TO (9).

d	Leading vehicle decelerating	Cut-in
1	36.9 %	36.9 %
2	63.0 %	63.3 %
3	78.0 %	84.5 %
4	90.4 %	94.1 %
5	94.5 %	96.9 %
6	96.7 %	99.0 %
7	98.2 %	99.6 %
8	99.2 %	99.8 %

Another way to determine d is to use the SR metric $M_p(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ defined in Section IV-C. In Fig. 7, the result is shown when applying this metric with $p = 1$, alongside with the empirical Wasserstein metric $\tilde{W}_1(\mathcal{Z}, \mathcal{W})$ and the penalty $\tilde{W}_1(\mathcal{Z}, \mathcal{W}) - \tilde{W}_1(\mathcal{X}, \mathcal{W})$. Each point in Fig. 7 represents the median[3] when applying the metric 200 times, each time with a different (random) partition of the training data $\mathcal{X}$ and test data $\mathcal{Z}$. The standard deviation of the medians in Fig. 7, estimated using bootstrapping [50], is 0.005 or less. For the

[3]We preferred to use the median instead of the mean, such that the result is less influenced by outliers [49].

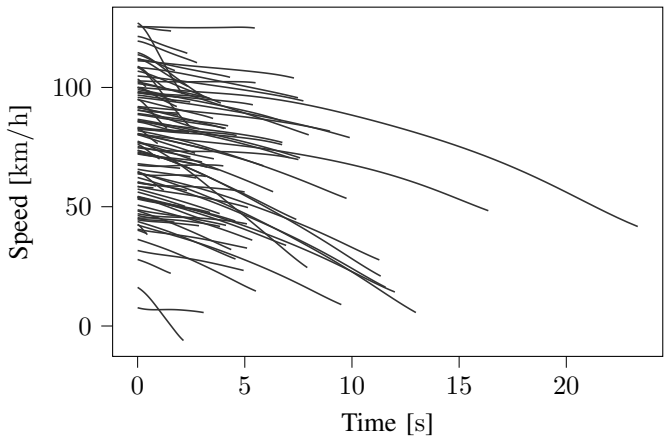

Fig. 6. Speed of the leading vehicle during 100 generated LVD scenarios.

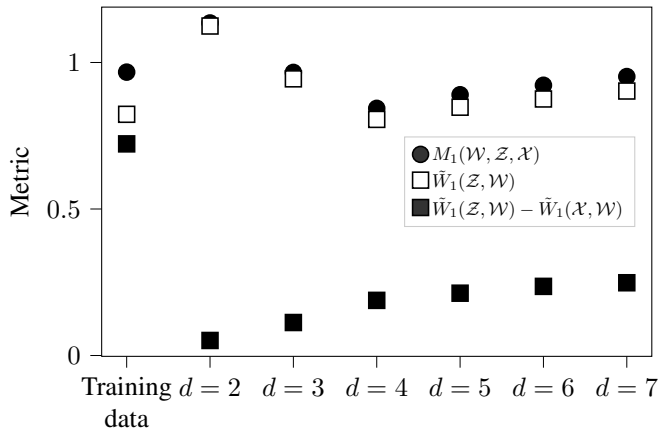

Fig. 7. Medians of the metrics for the set of generated LVD scenario parameters. Note that $d = 1$ is excluded, because its metrics are an order of magnitude higher than for $d = 2$ and would, therefore, not be visible with the current scaling of the y-axis.

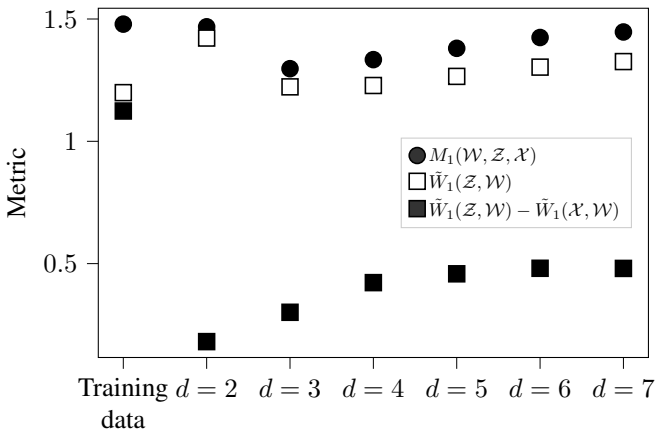

Fig. 8. Medians of the metrics for the set of generated cut-in scenario parameters.

SR metric, the penalty is weighted using $\beta = 0.25$. The choice of $\beta = 0.25$ is justified in Section V-E.

The most left points in Fig. 7 represent the metric in case the set of training data $\mathcal{X}$ is directly used to sample the scenario parameters instead of the approach of Section III. Here, $\mathcal{W}$ is a selection with replacement of N_w scenarios from $\mathcal{X}$, i.e.:

$$\mathbf{w}_i = \mathbf{x}_{\lfloor u \rfloor}, u \sim U(1, N_x + 1), \forall i \in \{1, \ldots, N_w\}, \qquad (24)$$

where $U(1, N_x + 1)$ denotes the continuous uniform distribution with boundaries 1 and $N_x + 1$, and $\lfloor \cdot \rfloor$ denotes the floor function. Using the training data directly for "generating scenarios" leads to a low empirical Wasserstein metric. The downside is that there is not much variation among the generated scenarios. Therefore, the penalty is also the highest, which results in $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X}) \approx 0.967$. Looking at $d = 4$, the empirical Wasserstein metric (open squares) is approximately similar compared to when the training set is directly used. Due to the sampling of the scenario parameters from the KDE, the generated scenarios contain more variation than the training set, resulting in a lower penalty and, therefore, a lower metric evaluation of $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X}) \approx 0.843$. Increasing d even further results in higher metric evaluations. So based on the proposed metric, $d = 4$ seems the right choice.

Figure 8 shows the results of the generation of the cut-in scenario parameters in a similar way as Fig. 7. The standard deviation of all points in Fig. 8 is less than 0.008. The lowest penalty is obtained with $d = 2$, but the higher empirical Wasserstein distance suggests that too much information is lost. The best result, i.e., where the SR metric, $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$, is minimal, is obtained at $d = 3$.

D. Comparison with other approaches

Our proposed method utilizes an SVD to obtain the scenario parameters and multivariate KDE to estimate the pdf of these parameters. To illustrate the advantages of these choices, the results of our method are compared with alternative approaches. First, instead of using an SVD for obtaining the parameters, a fixed parameterization is used, such as in [10, 18, 21]. Second, instead of using KDE to estimate the pdf of the parameters, a Gaussian distribution like in [29] is assumed. Third, the parameters are assumed to be independent.

When using a fixed parameterization for the LVD scenario, 4 parameters describe the scenario [10]: the speed reduction of the leading vehicle, the final speed of the leading vehicle, the duration of the scenario, and the initial time gap between the leading vehicle and the ego vehicle. The speed of the leading vehicle is assumed to follow a sinusoidal function, such that the acceleration at the start and at the end of the scenario equals zero. In case of the cut-in scenario, 5 parameters describe the scenario: the mean speed of the vehicle cutting in, its initial lateral position with respect to the center of the ego vehicle's lane, the duration of the scenario, the initial speed of the ego vehicle, and the initial longitudinal position of the vehicle cutting in with respect to the ego vehicle. The speed of the vehicle cutting in is assumed to be constant. Its lateral position is assumed to follow a sinusoidal function, such that the vehicle ends at the center of the ego vehicle's lane. For estimating the pdf of these parameters, the comparison considers 4 possibilities: multivariate KDE, multiple univariate KDEs, a multivariate Gaussian distribution, and multiple univariate Gaussian distributions.

Table IV shows the results of the different approaches for generating scenario parameters. For the LVD scenarios, our proposed approach (top row in Table IV) resulted in the lowest $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$. For the cut-in scenarios, it is interesting to note that the scores are not very different if SVD is used to obtain the parameters. This is partly explained by the smaller data set, because this results in a higher bandwidth[4] that makes the KDE result with the Gaussian kernel look more like a Gaussian distribution. Using SVD and KDE while assuming that the parameters are independent, results in an even better result:

[4]On average, the bandwidth is about 1.5 to 2 times larger for the cut-in scenarios compared to the LVD scenarios.

TABLE IV
MEDIANS OF THE METRIC $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ WITH DIFFERENT APPROACHES FOR GENERATING SCENARIOS.

Parameters	Distribution	Dependency	LVD	Cut-in
SVD	KDE	Dependent	0.84	1.30
SVD	Gaussian	Dependent	1.00	1.33
SVD	KDE	Independent	0.99	1.28
SVD	Gaussian	Independent	1.00	1.33
Fixed	KDE	Dependent	2.65	1.70
Fixed	Gaussian	Dependent	2.58	1.71
Fixed	KDE	Independent	2.31	1.67
Fixed	Gaussian	Independent	4.76	1.69

1.28 instead of 1.30 (with a standard deviation of 0.005). This indicates that assuming that the 3 parameters obtained with the SVD are independent, is acceptable.

E. Evaluating the scenario representativeness metric

To determine whether our proposed metric Section IV-C correlates better with the Wasserstein metric (14) than the empirical Wasserstein metric (15), the Wasserstein metric (14) needs to be known. This is not possible because the true underlying distribution of the data is unknown. To estimate the Wasserstein metric (14), the empirical Wasserstein metric (15) can be used with large numbers of test scenarios and generated scenario parameters, i.e., with large values of N_z and N_w, respectively. Since a large number of test scenarios is not available to us, we assume a certain distribution for $f(\cdot)$ from which the training data and the test data are generated. The approach is as follows (the numbers are for the LVD scenarios and, in parenthesis, the numbers for the cut-in scenarios are shown):

1) Based on the original 1150 (289) scenarios, obtained from the data, the following sets of scenario parameters are generated using the proposed approach explained in Section III with $d = 4$ ($d = 3$):
   - A new set of training data $\mathcal{X}^*$ of size $N_x = 920$ ($N_x = 231$);
   - A new set of test data $\mathcal{Z}^*$ of size $N_z = 230$ ($N_z = 58$); and
   - A large set of test data $\mathcal{Z}^*_{\text{large}}$ of size $N_z = 10000$ ($N_z = 10000$).
2) Based on $\mathcal{X}^*$, $N_w = 10000$ ($N_w = 10000$) scenario parameters are generated and collected in a set $\mathcal{W}^*$.
3) Our proposed metric is computed using $\mathcal{W}^*$, $\mathcal{Z}^*$, and $\mathcal{X}^*$: $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ with $\beta = 0.25$.
4) The Wasserstein metric of (14) is estimated using the empirical Wasserstein metric of (17) with $\mathcal{W}^*$ and $\mathcal{Z}^*_{\text{large}}$. Note: to approximate the Wasserstein metric of (14) using the empirical Wasserstein metric of (17), both $\mathcal{W}^*$ and $\mathcal{Z}^*_{\text{large}}$ need to be large (but not necessarily the same) in size.

We have repeated this approach 200 times, each time with a different (random) partition of the training data $\mathcal{X}$ and test data $\mathcal{Z}$. Figs. 9 and 10 show the result of this approach for the LVD scenarios and cut-in scenarios, respectively. In both cases, the

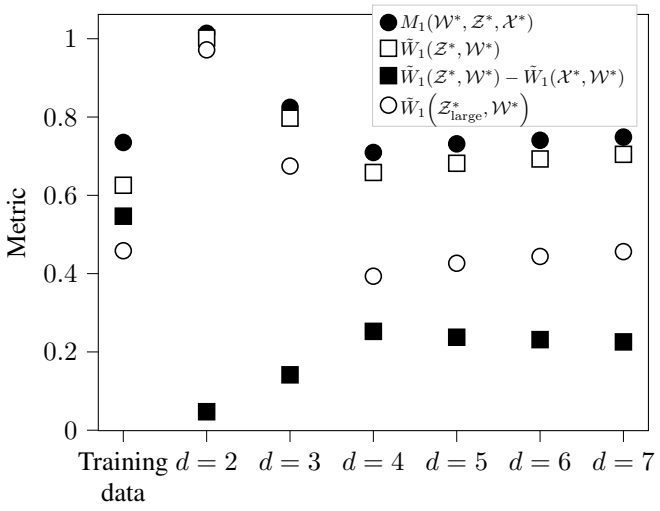

Fig. 9. Medians of the metrics for the set of $N_w = 10000$ generated LVD scenario parameter vectors. In this case, the $N_x = 920$ scenarios of $\mathcal{X}^*$ are sampled from $\hat{f}_H(\cdot)$ of (11), where $\hat{f}_H(\cdot)$ is based on the original data set $\mathcal{X}$.

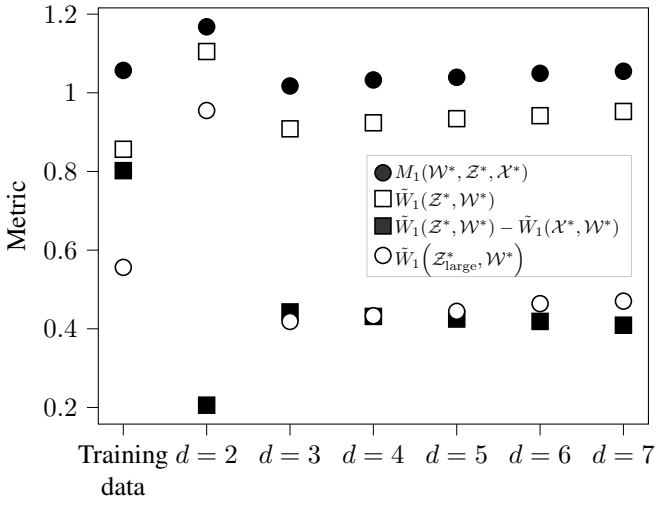

Fig. 10. Medians of the metrics for the set of $N_w = 10000$ generated cut-in scenario parameter vectors.

empirical Wasserstein metric $\tilde{W}_1(\mathcal{Z}^*, \mathcal{W}^*)$ is minimal when the training data are directly used for the generated scenario parameters. Thus, the empirical Wasserstein metric suggests that the best approach for generating new scenario parameters is to simply sample parameters from the training data. The actual Wasserstein metric, estimated using $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ shows that using our proposed method outperforms sampling parameters directly from the training data.

To justify the choice of $\beta = 0.25$, Fig. 11 shows the correlation between the medians of the proposed metric $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ for different values of β. With $\beta = 0$, i.e., $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*) = \tilde{W}_1(\mathcal{Z}^*, \mathcal{W}^*)$, the correlation is 0.974 for the LVD scenarios and 0.824 for

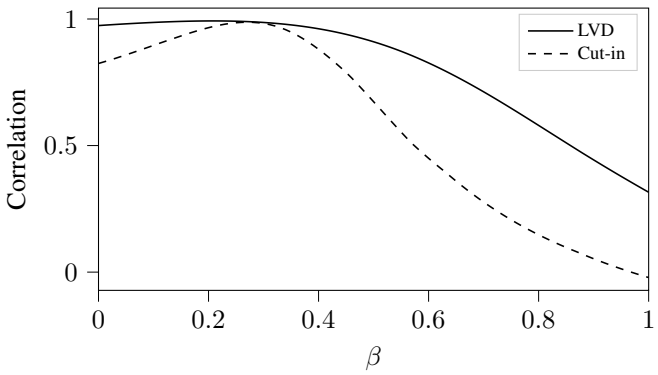

Fig. 11. Correlation between the medians of $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ (open circles in Figs. 9 and 10) and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ (filled circles in Figs. 9 and 10) for different values of β. The solid line shows the result for the LVD scenarios with a maximum correlation of 0.992 at $\beta \approx 0.21$. The dashed line shows the result for the cut-in scenarios with a maximum correlation of 0.987 at $\beta \approx 0.27$.

the cut-in scenarios. The correlation increases with increasing β until the maximum is obtained at $\beta \approx 0.21$ for the LVD scenarios and at $\beta \approx 0.27$ for the cut-in scenarios. The correlations at these maxima are 0.992 and 0.987, respectively. Increasing β further results in a lower correlation, which suggests that a choice of $\beta = 0.25$ seems appropriate.

The experiment described in this section can be used to determine both d and β in an iterative manner given an initial choice for β (denoted by β_0):

1) Set $i = 0$.
2) Determine d_i, i.e., the optimal number of parameters that minimizes $M_1(\mathcal{W}, \mathcal{Z}, \mathcal{X})$ using $\beta = \beta_i$.
3) Generate $\mathcal{X}^*$, $\mathcal{W}^*$, $\mathcal{Z}^*$, and $\mathcal{Z}^*_{\text{large}}$ using the approach described in this section with $d = d_i$.
4) Increase i by 1.
5) Determine β_i by maximizing the correlation between $M_1(\mathcal{W}^*, \mathcal{Z}^*, \mathcal{X}^*)$ with $\beta = \beta_i$ and $\tilde{W}_1\left(\mathcal{Z}^*_{\text{large}}, \mathcal{W}^*\right)$ (e.g., see Fig. 11).
6) Repeat step 2.
7) Stop if $d_i = d_{i-1}$. Otherwise, return to step 3.

As an initial choice, $\beta_0 = 0.25$ seems appropriate. More specifically, when choosing $\beta_0 \in [0.1, 1]$, the optimal choice of d is found after one iteration.

VI. Discussion

One of the advantages of our proposed method for generating scenario parameters is that less assumptions are needed regarding the parameterization of the scenarios:

- There is no assumption needed on a predetermined functional form of the time series data. For example, in an LVD scenario, the speed is often assumed to follow a polynomial function [10], a sinusoidal function, or a linear function [21]. In case of a predetermined functional form, parameters are fitted to the functional form. In our case, the SVD automatically determines the optimal choice of parameterization without relying on a predetermined functional form.
- There is no assumption needed for the shape of the distribution of the parameters. For example, a particular distribution, such as a Gaussian distribution [29] or a uniform distribution, may be assumed for which parameters are fitted. Alternatively, assumptions are made regarding the independence of the parameters [24]. In our case, the KDE automatically adapts its shape to the data and also considers the dependence among the different parameters.

It should be noted, however, that if there is a reason to believe that one or more of the assumptions are valid, than alternative methods for generation scenario parameters that make use of such assumptions might perform equally or better than the presented method [51]. In most cases, it will be difficult to provide a proper justification of the assumptions regarding the functional form of, e.g., the vehicle speed, and the pdf of the scenario parameters and the presented method will outperform methods relying on such assumptions. In any case, the presented SR metric provides an opportunity to verify the applicability of any assumptions regarding the scenario parameterization and parameter distributions.

The generated scenario parameters represent scenarios that could happen in real life and cover the same variety that is found in real-world traffic. Most likely, the majority of these scenarios are straightforward for the AV to deal with. To do an efficient assessment, the focus should be on scenarios that might lead to critical situations in which the probability of collision is high. That is why so-called *importance sampling* [52, Chapter 5.6] is often used for the assessment of AVs, e.g., see [5, 10, 53, 54]. With importance sampling, a different pdf, $g(\cdot)$, is used to sample scenario parameters, such that more emphasis is put on scenarios that might lead to critical situations. To get unbiased results, the result of a test with scenario parameters $\mathbf{x}$ is weighted by the ratio of the original probability density, $\hat{f}_H(\mathbf{x})$, and the probability density of the pdf used for importance sampling, $g(\mathbf{x})$ [10, 52, 53, 54]. Note that the importance sampling techniques explained in [10, 53, 54] can be directly applied on the estimated pdf $\hat{f}_H(\cdot)$ in (11) of the reduced set of parameters. In future work, our method for generating scenarios will be combined with importance sampling [10, 53, 54] for an assessment of an AV.

In some cases, one might want to sample from a conditional pdf, e.g., in case of sampling the scenario parameters for the LVD scenario such that the initial time gap equals a specified value. Sampling from a KDE such that one or more parameters are predetermined is straightforward [55]. In our case, sampling from $\hat{f}_h(\cdot)$ such that the time gap equals a specified value results in a linear constraint on the samples, because the reduced parameter vector $\tilde{\mathbf{v}}_i$ of (10) results from a linear mapping of the original parameters $\mathbf{x}_i$ of (2). In other words, one might want to sample v from $\hat{f}_H(\cdot)$ of (11), such that v is subject to the linear constraint

$$Av = b, \tag{25}$$

where A and b are a matrix and vector, respectively. In [40], an algorithm is provided for sampling from a pdf estimated

using KDE such that the generated sample is subject to the constraint of (25). The main idea of [40] is to weight each parameter vector v_i, $i \in \{1, \ldots, N_x\}$ in the KDE based on how closely the v_i matches the constraint (25).

The presented case study considers a vehicle for which the full trajectory is predetermined. For the presented scenarios, this works well, but the full trajectory is not predetermined in scenarios where the actor's behavior depends on the behavior of the ego vehicle [56]. To deal with such scenarios, one option is to use a driver behavior model (e.g., [57, 58]) with predefined parameters instead of describing the full trajectory. The parameters of the driver behavior model may be part of θ. The proposed method for generating scenario parameter values still applies in these kind of scenarios. Our ongoing research focuses on the assessment of AVs using scenarios in which driver behavior models are used for vehicles that may respond to the ego vehicle's behavior.

Since KDE is used, the generated scenario parameters represent variations of the data. Nevertheless, if the data do not contain scenarios that might lead to critical situations, such as an emergency braking maneuver or a reckless cut-in scenario, it is unlikely that such scenarios are generated, even if importance sampling [10, 53, 54] is used. Therefore, when using the generated scenarios for the (safety) assessment of AVs, it is important that there is enough data such that the data contain such scenarios. Although there is no consensus yet on the required amount of data, some metrics have been proposed [39, 59] for determining whether enough data have been collected when using the data for the assessment of AVs.

This work employs the Wasserstein metric to propose the SR metric for evaluating the generated scenario parameters. It is illustrated how our proposed metric could be used to determine the appropriate number of parameters (d) and the type of distribution that is used to model the pdf of the scenario parameters. Also, the bandwidth h or bandwidth matrix H could also be determined by optimizing the proposed metric. In case of the bandwidth estimation, the disadvantage is that it would require more computational resources compared to, e.g., leave-one-out cross-validation.

More research is needed to determine the influences on the optimal choice for the penalty weight β. The case study has demonstrated one way to verify whether the initial choice of β was appropriate, but we do not yet know *why* a weight of $\beta \approx 0.25$ is an appropriate choice. The actual choice might depend on, among others, N_x, N_z, N_w, and the shape of the underlying distribution of the scenario parameters. Future research with a larger data set will allow us to better determine the optimal β and how this optimal value is influenced.

Future work involves researching the use of the proposed metric in combination with alternative methods for generating scenarios for the assessment of AVs. For example, Spooner et al. [28] have used a GAN [60] to create pedestrian crossing scenarios. One of the difficulties with GANs is to know when the GAN truly replicates the underlying distribution. Several metrics have been proposed [61] to evaluate the performance of GANs, among which a metric based on the Wasserstein metric that compares the generated data with test data. Alternatively, our proposed metric, which also considers the training data, could be considered for evaluating GANs. To judge the potential of our proposed metric in this application, more research is needed.

VII. Conclusions

It is essential for the deployment of cro:avAutomated Vehicles (AVs) to develop assessment methods. Scenario-based assessment in which test cases are derived from real-world road traffic scenarios is regarded as a viable approach for assessing AVs. This work has presented a method to generate parameterized scenarios for the use in test case descriptions for the assessment of AVs. To not rely on a small set of parameters, we have used cro:svdSingular Value Decomposition (SVD) to reduce the parameters. Parameter values for the scenarios are generated by drawing samples from the estimated cro:pdfprobability density function (pdf) of the reduced set of parameters. To deal with the unknown shape of the pdf, it has been proposed to estimate the pdf using cro:kdeKernel Density Estimation (KDE). This work has also presented a novel metric, the so-called cro:srScenario Representativeness (SR) metric, based on the Wasserstein metric, for evaluating whether the generated scenario parameters represent realistic scenarios while covering the same variety that is found in real-world traffic.

A case study has illustrated the proposed method for generating scenario parameter values using scenarios with a leading vehicle that decelerates and scenarios with a vehicle that performs a cut-in. The case study has also illustrated that the proposed metric correctly quantifies the degree to which the generated scenario parameter values represent real-world scenarios and, at the same time, cover the same variety of scenarios that is found in real life.

Future work involves applying the proposed method for more complex scenarios, e.g., scenarios that contain several different actors, to generate scenario-based test cases for the safety assessment of AVs. Additionally, it would be of interest to apply importance sampling for AV assessment in combination with the proposed method for generating scenarios. Other future work involves investigating the use of the proposed metric in combination with alternative methods for generating scenarios for the assessment of AVs.

References

[1] K. Bengler, K. Dietmayer, B. Färber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[2] J. E. Stellet, M. R. Zofka, J. Schumacher, T. Schamm, F. Niewels, and J. M. Zöllner, "Testing of advanced driver assistance towards automated driving: A survey and taxonomy on existing approaches and open questions," in *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015, pp. 1455–1462.

[3] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, pp. 15–24, 2016.

[4] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[5] D. Zhao, X. Huang, H. Peng, H. Lam, and D. J. LeBlanc, "Accelerated evaluation of automated vehicles in car-following maneuvers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 733–744, 2018.

[6] S. Riedmaier, T. Ponn, D. Ludwig, B. Schick, and F. Diermeyer, "Survey on scenario-based safety assessment of automated vehicles," *IEEE Access*, vol. 8, pp. 87 456–87 477, 2020.

[7] H. Elrofai, J.-P. Paardekooper, E. de Gelder, S. Kalisvaart, and O. Op den Camp, "Scenario-based safety validation of connected and automated driving," Netherlands Organization for Applied Scientific Research, TNO, Tech. Rep., 2018.

[8] A. Pütz, A. Zlocki, J. Bock, and L. Eckstein, "System validation of highly automated vehicles with a database of relevant traffic scenarios," in *12th ITS European Congress*, 2017, pp. 1–8.

[9] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *IEEE 21st International Conference on Intelligent Transportations Systems (ITSC)*, 2018, pp. 2118–2125.

[10] E. de Gelder and J.-P. Paardekooper, "Assessment of automated driving systems using real-life scenarios," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 589–594.

[11] J. Antona-Makoshi, N. Uchida, K. Yamazaki, K. Ozawa, E. Kitahara, and S. Taniguchi, "Development of a safety assurance process for autonomous vehicles in Japan," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019, pp. 1–18.

[12] G. H. Golub and C. F. Van Loan, *Matrix Computations*. John Hopkins University Press, 2013, vol. 3.

[13] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, vol. 27, no. 3, pp. 832–837, 1956.

[14] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[15] L. Rüschendorf, "The Wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985.

[16] L. Li, W.-L. Huang, Y. Liu, N.-N. Zheng, and F.-Y. Wang, "Intelligence testing for autonomous vehicles: A new approach," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 2, pp. 158–166, 2016.

[17] U. Lages, M. Spencer, and R. Katz, "Automatic scenario generation based on laserscanner reference data and advanced offline processing," in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 2013, pp. 146–148.

[18] M. R. Zofka, F. Kuhnt, R. Kohlhaas, C. Rist, T. Schamm, and J. M. Zöllner, "Data-driven simulation and parametrization of traffic scenarios for the development of advanced driver assistance systems," in *18th International Conference on Information Fusion*, 2015, pp. 1422–1428.

[19] L. Stepien, S. Thal, R. Henze, H. Nakamura, J. Antona-Makoshi, N. Uchida, and P. Raksincharoensak, "Applying heuristics to generate test cases for automated driving safety evaluation," *Applied Sciences*, vol. 11, no. 21, p. 10 166, 2021.

[20] S. Feng, Y. Feng, C. Yu, Y. Zhang, and H. X. Liu, "Testing scenario library generation for connected and automated vehicles, part I: Methodology," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1573–1582, 2020.

[21] S. Thal, H. Znamiec, R. Henze, H. Nakamura, H. Imanaga, J. Antona-Makoshi, N. Uchida, and S. Taniguchi, "Incorporating safety relevance and realistic parameter combinations in test-case generation for automated driving safety assessment," in *IEEE Intelligent Transportation Systems Conference (ITSC)*, 2020, pp. 666–671.

[22] S. Feng, Y. Feng, X. Yan, S. Shen, S. Xu, and H. X. Liu, "Safety assessment of highly automated driving systems in test tracks: A new framework," *Accident Analysis & Prevention*, vol. 144, p. 105 664, 2020.

[23] L. Li, N. Zheng, and F.-Y. Wang, "A theoretical foundation of intelligence testing and its application for intelligent vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 6297–6306, 2020.

[24] S. Feng, X. Yan, H. Sun, Y. Feng, and H. X. Liu, "Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment," *Nature Communications*, vol. 12, no. 748, pp. 1–14, 2021.

[25] M. Koren, S. Alsaif, R. Lee, and M. J. Kochenderfer, "Adaptive stress testing for autonomous vehicles," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1898–1904.

[26] A. Corso and M. J. Kochenderfer, "Interpretable safety validation for autonomous vehicles," in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[27] F. Schuldt, A. Reschka, and M. Maurer, "A method for an efficient, systematic test case generation for advanced driver assistance systems in virtual environments," in *Automotive Systems Engineering II*, Springer, 2018, pp. 147–175.

[28] J. Spooner, V. Palade, M. Cheah, S. Kanarachos, and A. Daneshkhah, "Generation of pedestrian crossing scenarios using ped-cross generative adversarial network," *Applied Sciences*, vol. 11, no. 2, p. 471, 2021.

[29] O. Gietelink, "Design and validation of advanced driver assistance systems," Ph.D. dissertation, Delft University of Technology, 2007.

[30] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[31] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[32] D. Zhao, H. Lam, H. Peng, S. Bao, D. J. LeBlanc, K. Nobukawa, and C. S. Pan, "Accelerated evaluation of automated vehicles safety in lane-change scenarios based on importance sampling techniques," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 595–607, 2016.

[33] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, 1992.

[34] E. de Gelder, J.-P. Paardekooper, A. Khabbaz Saberi, H. Elrofai, O. Op den Camp, S. Kraines, J. Ploeg, and B. De Schutter, "Towards an ontology for scenario definition for the assessment of automated vehicles: An object-oriented framework," *In preparation*, 2022.

[35] C. de Boor, *A Practical Guide to Splines*. Springer, 1978, vol. 27.

[36] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[37] B. A. Turlach, "Bandwidth selection in kernel density estimation: A review," Institut für Statistik und Ökonometrie, Humboldt-Universität zu Berlin, Tech. Rep., 1993.

[38] T. Duong, "ks: Kernel density estimation and kernel discriminant analysis for multivariate data in R," *Journal of Statistical Software*, vol. 21, no. 7, pp. 1–16, 2007.

[39] E. de Gelder, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Safety assessment of automated vehicles: How to determine whether we have collected enough field data?" *Traffic Injury Prevention*, vol. 20, no. S1, pp. 162–170, 2019.

[40] E. de Gelder, E. Cator, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Constrained sampling from a kernel density estimator to generate scenarios for the assessment of automated vehicles," in *IEEE Intelligent Vehicles Symposium Workshops (IV Workshop)*, 2021, pp. 203–208.

[41] R. P. W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Transactions on Computers*, vol. C-25, no. 11, pp. 1175–1179, 1976.

[42] A. Z. Zambom and R. Dias, "A review of kernel density estimation with applications to econometrics," *International Econometric Review (IER)*, vol. 5, no. 1, pp. 20–42, 2013.

[43] A. Gramacki, *Nonparametric Kernel Density Estimation and Its Computational Aspects*, J. Kacprzyk, Ed. Springer, 2018.

[44] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[45] M. Sommerfeld and A. Munk, "Inference for empirical Wasserstein distances on finite spaces," *Journal of the Royal Statistical Society Series B*, vol. 80, no. 1, pp. 219–238, 2018.

[46] J.-P. Paardekooper, S. Montfort, J. Manders, J. Goos, E. de Gelder, O. Op den Camp, A. Bracquemond, and G. Thiolon, "Automatic identification of critical scenarios in a public dataset of 6000 km of public-road driving," in *26th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2019.

[47] J. Elfring, R. Appeldoorn, S. van den Dries, and M. Kwakkernaat, "Effective world modeling: Multisensor data fusion methodology for automated driving," *Sensors*, vol. 16, no. 10, pp. 1–27, 2016.

[48] E. de Gelder, J. Manders, C. Grappiolo, J.-P. Paardekooper, O. Op den Camp, and B. De Schutter, "Real-world scenario mining for the assessment of automated vehicles," in *IEEE International Transportation Systems Conference (ITSC)*, 2020, pp. 1073–1080.

[49] B. Doerr and A. M. Sutton, "When resampling to cope with noise, use median, not mean," in *Genetic and Evolutionary Computation Conference*, 2019, pp. 242–248.
[50] B. Efron, "Bootstrap methods: Another look at the jackknife," in *Breakthroughs in Statistics*, Springer, 1992, pp. 569–593.
[51] S. Siegel, "Nonparametric statistics," *The American Statistician*, vol. 11, no. 3, pp. 13–19, 1957.
[52] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*. John Wiley & Sons, 2016.
[53] S. Jesenski, N. Tiemann, J. E. Stellet, and J. M. Zöllner, "Scalable generation of statistical evidence for the safety of automated vehicles by the use of importance sampling," in *IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–8.
[54] Y. Xu, Y. Zou, and J. Sun, "Accelerated testing for automated vehicles safety evaluation in cut-in scenarios based on importance sampling, genetic algorithm and simulation applications," *Journal of Intelligent and Connected Vehicles*, vol. 1, pp. 28–38, 2018.
[55] M. P. Holmes, A. G. Gray, and C. L. Isbell, "Fast nonparametric conditional density estimation," in *23rd Conference on Uncertainty in Artificial Intelligence*, 2007, pp. 175–182.
[56] M. Althoff, M. Koschi, and S. Manzinger, "CommonRoad: Composable benchmarks for motion planning on roads," in *IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 719–726.
[57] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical review E*, vol. 62, no. 2, pp. 1805–1824, 2000.
[58] A. Kesting, M. Treiber, and D. Helbing, "General lane-changing model MOBIL for car-following models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, 2007.
[59] W. Wang, C. Liu, and D. Zhao, "How much data are enough? A statistical approach with case study on longitudinal driving behavior," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 2, pp. 85–98, 2017.
[60] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *27th International Conference on Neural Information Processing Systems*, vol. 2, 2014, pp. 2672–2680.
[61] A. Borji, "Pros and cons of GAN evaluation measures," *Computer Vision and Image Understanding*, vol. 179, pp. 41–65, 2019.