Processing, assessing, and enhancing the Waymo autonomous vehicle open dataset for driving behavior research

Xiangwang Hu[a,b]; Zuduo Zheng[a*]; Danjue Chen[c]; Xi Zhang[c]; Jian Sun[b]

[a]School of Civil Engineering, The University of Queensland, Australia
[b]Department of Traffic Engineering, Tongji University, China
[c]Civil and Environmental Engineering, University of Massachusetts, Lowell, United States

Abstract: Recently released Autonomous Vehicle (AV) trajectory datasets can potentially catalyze research progress on AV-oriented traffic flow analysis. This paper aims to comprehensively and systematically process and assess one of the AV-oriented open datasets, i.e., Waymo Open Dataset, with a focus on car following paired trajectories. First, the original dataset has been processed into a user-friendly format which contains all important information related to the behavior of AV and surrounding objects. Second, the data quality has been assessed in terms of internal consistency, jerk values and trajectory completeness. Results show that the extracted trajectories are all incomplete but generally they have better quality than that of Next Generation Simulation program (NGSIM) dataset. Third, the trajectory data has been further enhanced by using an optimization-based outlier removal method and a wavelet denoising method. Additionally, we have tested the impact of data outliers and noise on IDM calibration, and revealed significant differences in parameter values for desired time gap T and maximum acceleration a.

Keywords: autonomous vehicle; trajectory data; outlier removal; denoising; driving behavior; car following

1. Introduction

Trajectory data play a critical role in traffic flow studies, microscopic modelling in particular (Li et al., 2020). In the past decades, thanks to the emergence of high-resolution and openly-accessible trajectory datasets, many traffic flow phenomena have been observed and studied using detailed empirical analysis, such as: traffic hysteresis (Yeo and Skabardonis, 2009, Tordeux et al., 2010, Huang et al., 2018, Laval, 2011, Chen et al., 2012b, Saifuzzaman et al., 2017), traffic oscillations (Zheng et al., 2011b, Chen et al., 2012a, Chen et al., 2014, Tian et al., 2016), heterogeneity (Ossen and Hoogendoorn, 2011, Moridpour et al., 2015); and numerous models have been proposed to better approximate car following behavior (Ahn et al., 2004, Colombaroni and Fusco, 2013, Laval et al., 2014, Saifuzzaman and Zheng, 2014, He et al., 2015, Sharma et al., 2019b) and lane changing behavior(Leclercq et al., 2007, Thiemann et al., 2008, Zheng et al., 2013, Yi et al., 2014, Zheng, 2014, Ali et al., 2020b, Ali et al., 2020a) .

Traditionally, trajectory data are collected using image processing method based on recorded videos from either fixed cameras or drones. The most celebrated trajectory dataset is perhaps the Next Generation Simulation program (NGSIM) dataset (NGSIM,

* Corresponding author. Email: zuduo.zheng@uq.edu.au.

2016) which has a total duration of 150 minutes from fixed cameras at 4 sites (2 from highways and 2 from urban streets). Another popular dataset is highD Dataset collected by camera-equipped drones, which has a total duration of 16.5 hours at 6 locations of German highways (Krajewski et al., 2018). Using the same method, Bock et al. (2019) collected and released the intersection counterpart dataset which is called inD Dataset. Recently, several new datasets pertaining to vehicles at high-level automation (Level 4 according to NHTSA (2021), referred to as autonomous vehicles (AV), have been released such as KITTI (Geiger et al., 2013), Argo Dataset (Chang et al., 2019), Lyft Level 5 AV Dataset (Kesten et al., 2019), BDD100K (Yu et al., 2020), nuScenes Dataset (Caesar et al., 2020) and Waymo Open Dataset (Sun et al., 2020). Different from the trajectory datasets for traditional vehicles, these datasets are usually collected by onboard sensors. Apart from camera images, they often also contain Lidar information which can produce 3D object bounding box for each object, and trajectory data can be obtained by continuously tracking objects. Among these datasets related to AV, Lyft Level 5 AV Dataset, nuScenes Dataset and Waymo Open Dataset collected trajectories of the AV and human-driven vehicles (referred to as HV hereafter) from real world traffic. These datasets are referred to as the AV-oriented empirical datasets. In these datasets, both the information on the movement of AV itself and the information on AV's surrounding environment are detected and extracted. Thus, these three datasets are particularly useful for driving behavior research. An overview of these datasets is given in Table 1. More data are likely to be released by these companies in the future.

Table 1. Overview of three AV trajectory dataset

Dataset	Number of segments	Resolution (s)	Length of each segment (s)
Waymo	1000	0.1	20
Lyft	366	0.2	25-45
nuScenes	1000	0.5	20

Although it is widely speculated that AV is likely to revolutionize road transportation systems, more and more researchers have cautioned that in transitioning to AV, HV and AV will have to co-exist for a considerable amount of time(Sharma and Zheng, 2021). Understanding interactions between HV and AV and the resulting traffic dynamics is critical for materializing the often-discussed benefits of AV, such as improving traffic safety, reducing traffic congestion, reducing energy consumption and vehicle emission, etc. Unfortunately, due to the lack of large-scale empirical data of mixed traffic, most studies on impact of AV on traffic either rely on numerical simulations or data from driving simulators (Ali et al., 2019, Sharma et al., 2019c), which can put the reliability of their conclusions into question. Obviously, the newly released AV-oriented empirical datasets can fill this gap. Using the high-resolution field observations contained in these datasets, researchers can realistically and reliably investigate the behavior of AV and its impact on traffic flow, and other related issues, such as the interaction between AV and cyclists or pedestrians, AV's impact on energy consumption, emissions, etc.

However, these AV-oriented empirical datasets can be difficult for traffic flow researchers to understand and use. First, these datasets were collected by an array of sensors, some of which (e.g., Lidar) are new to researchers in the traffic flow community. Second, the information collected by these sensors is much more complicated than a typical dataset collected by traffic flow research community, because it collected not only detailed information about the movement of AV, but also a huge amount of information of all the objects falling into its detection range. Finally, the structure and format of these dataset are complicated and not user friendly. For example, data from different sensors (e.g., camera, radar, Lidar) are often not integrated, but stored separately.

On top of these aforementioned factors, the data error and noise in these datasets are inevitable, and can influence the reliability of further analysis or modelling. Even for the widely-used NGSIM dataset, which focused on HV and was much simpler than the AV-oriented empirical datasets, significant errors and noise have been frequently studied and reported (Duret et al., 2008, Thiemann et al., 2008, Punzo et al., 2011, Montanino and Punzo, 2015, Coifman and Li, 2017).

Therefore, this study aims to comprehensively and systematically process and assess one of the AV-oriented empirical datasets, i.e., Waymo Open Dataset in light of its abundant segments (driving scenarios) and high resolution, with a focus on car following paired trajectories. The same method can be applied to other AV-oriented empirical datasets. Our effort consists of three major components: data processing, quality assessing, and quality enhancing. Data processing includes trajectory extraction and visualization; quality assessing includes consistency analysis, jerk value analysis and trajectory completeness analysis; and quality enhancing includes outlier removal and denoising. In the processed Waymo Open Dataset, all important information related to driving behavior of AV and surrounding vehicles and other road users has been integrated into a single file in a format similar to NGSIM data, which is ready and easy to use for the transportation research community (the processed dataset is available from https://data.mendeley.com/datasets/wfn2c3437n/2). Moreover, the data quality is higher than the original because the outliers have been removed, noise has been filtered, its consistency has been checked, and the trajectory completeness has been analyzed.

The study was primarily motivated by contributing to the traffic flow community a NSGIM-like dataset for understanding interactions between HV and AV and the resulting traffic dynamics. We believe that, this easy-to-use, high-quality, and information-rich dataset for mixed traffic can potentially play an important role in the research of AV similar to the role of NGSIM, and catalyze research progress on AV's impact in mixed traffic. To better support other researchers in extracting their own trajectories, the data processing codes for this paper, together with two versions of the processed dataset, have been shared (see the Conclusions section). In addition, this paper has made a couple of methodological contributions as outlined below.

- We have developed a generally applicable and semi-automated procedure for processing AV-oriented empirical datasets, assessing and enhancing their quality. With the shared codes and shared two versions of our processed dataset, researchers can easily apply the same procedure to process other AV datasets.
- We have proposed a simple but effective, and optimization-based method for outlier removal. Also, we have used wavelet transform to filter trajectory data.

Remainder of this paper is organized as follows. Section 2 provides an overview of the Waymo Open Dataset; Section 3 introduces the data processing framework; Section 4 includes data processing, visualization and selection of CF vehicle pair. Data quality is assessed in Section 5 and further enhanced in Section 6. Finally, Section 7 summarizes the main conclusions.

2. The Waymo dataset

The Waymo Open Dataset consists of large-scale and high-resolution sensor data collected by Waymo autonomous vehicles in multiple cities in US (i.e., San Francisco, Phoenix, and Mountain View). A total of 1000 segments (scenarios) were originally released in 2019, and this number is continuously growing (1950 segments as of Nov 2020). The driving conditions covered in this dataset is diverse in terms of road types (urban streets, freeways, constructions), weather (sunny, rain), and time of day (dawn, day, dusk, night). The sensor data were collected by 5 Lidar (1 mid-range and 4 short-range) and 5 cameras (front and sides), where Lidar and camera were calibrated and synchronized. In addition, a large number of 3D ground truth bounding boxes (labels) for Lidar data were manually annotated for the purpose of object tracking. This dataset can be extremely valuable for the research community because of its large scale, diversities and reasonable quality.

Each file (file type '.tfrecord') downloaded from the Waymo Open Dataset website (https://waymo.com/open/download/) contains a number of segments (this research focuses on the original 1000 segments). Each segment has about 200 frames with a time interval of 0.1 s between two consecutive frames. Information in a single frame includes environment context, timestamp, AV's pose, camera images, camera labels, Lidar points, Lidar labels, etc. Among all the information provided, AV's pose, camera images and Lidar labels are extracted for the purpose of driving behavior research. More information on the Waymo Open Dataset can be found from this link: https://waymo.com/open/data/ .

3. The framework for data processing, assessing and enhancing

For traffic flow research (car following modelling in particular in this paper), we have the following goals for data processing, assessing and enhancing:

- Transforming the hierarchical data structure described above to a more user-friendly tabular data structure;
- Extracting the trajectories of both AV and its surrounding objects (including vehicles driven by human drivers, cyclists and pedestrians);
- For a better visual verification, generating videos (i) from the AV's perspective based on camera images; and (ii) from the top view perspective based on Lidar label positions;
- Selecting appropriate car-following (CF) vehicle pairs for CF behavior research;
- Assessing the quality of trajectory data; and
- Enhancing the data quality by removing outliers and denoising.

To achieve the above goals, a 3-stage data processing procedure has been implemented, as shown in the flowchart in Figure 1. Each stage is described in detail in Section 4, 5, and 6.

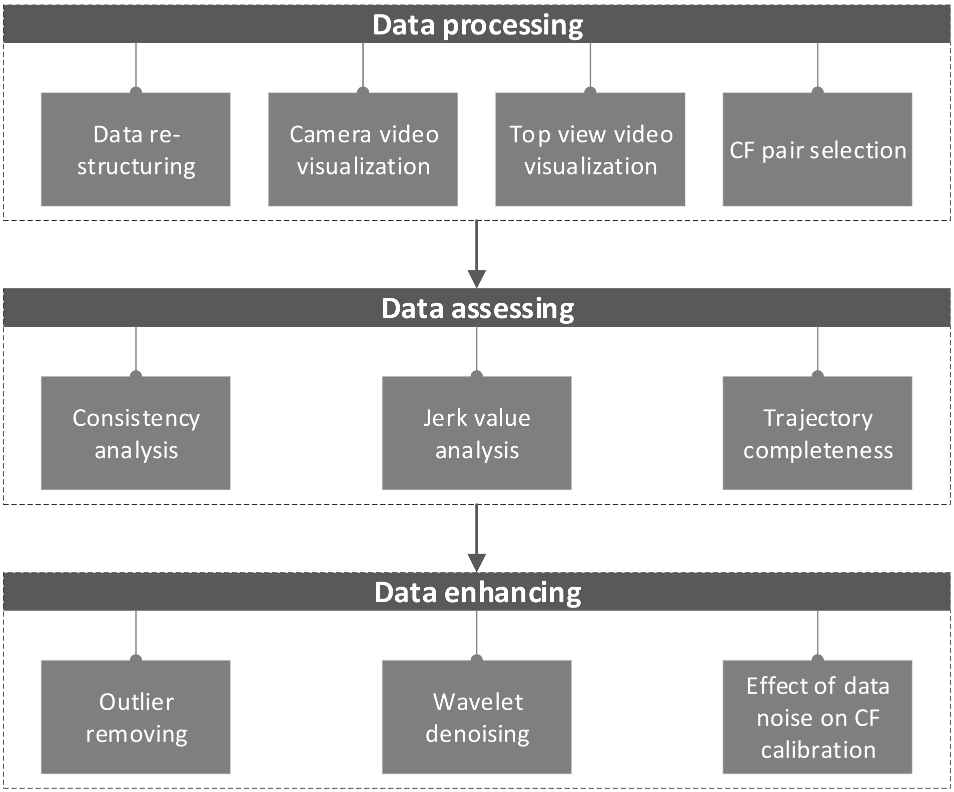


Figure 1. Flow chart of the data processing, assessing and enhancing procedure; CF: car following

4. Data processing

4.1 Information collection and re-structuring

To begin with, the original data are transformed from the original hierarchical structure to the tabular structure with 25 attributes. These attributes are related to frame context information (attributes 1-6), object characteristics (attributes 7-8 & 17-19) and object trajectory information (the rest), as shown in Table 2. Detailed description of each attribute is given in Appendix A.

Table 2. Attributes of the tabular structure

Attribute 1-6	Attribute 7-12	Attribute 13-18	Attribute 19-24	Attribute 25
'segment_id'	'obj_type'	'local_center_z'	'height'	'angular_speed'
'frame_label'	'obj_id'	'global_center_x'	'heading'	
'time_of_day'	'global_time_stamp'	'global_center_y'	'speed_x'	
'location'	'local_time_stamp'	'global_center_z'	'speed_y'	
'weather'	'local_center_x'	'length'	'accel_x'	
'laser_veh_count'	'local_center_y'	'width'	'accel_y'	

Note that in the original Waymo Open Dataset, the information of AV itself and the information of Lidar objects are stored separately. The trajectories of AV are in global coordinates, while the trajectories of Lidar objects are in local coordinates[†]. For traffic flow research, we need everything in a consistent context; i.e., either local or global. For AV, the global positions (center x, y, z) are directly extracted, while its local positions are always set to 0. On the other hand, for Lidar objects, their local positions are directly extracted, while their global positions are derived by using Equation (1):

$$p' = Ap \tag{1}$$

where p' is the global position, p is the local position, and A is the transformation matrix ('AV pose' in the original data).

As a side note, in our process a local integer number has been assigned to each segment as segment ID, replacing the original long and globally unique name for better readability.

4.2 Information visualization

To obtain a clear overview and visual verification of each segment, videos are generated in this stage, using camera images and Lidar information.

For the videos from camera images, they are simply a series of consecutive images at 10 frames per second. As an example, five images from five different directions (front, front left, side left, front right and side right) for Segment 391 at the same instant are presented together for visual verification, as shown in Figure 2.

Figure 2. Example: A camera video screenshot from Segment 391

For the Lidar information, we only show the top view (called trajectory view hereafter), from which trajectories can be directly observed. Specifically, the trajectory view videos display the real-time global positions of all the objects detected by Lidar at each time step, for which the object's global heading (defined as the angle of object's forward direction with respect to the global *x* direction) needs to be used. However, the headings for Lidar objects are measured in local coordinate, although the heading for

[†]According to Waymo, the global coordinates are 'East-North-Up' coordinates, and the local coordinates are related to the AV pose, where x-axis is positive forwards, y-axis is positive to the left, and z-axis is positive upwards.

AV is already in global coordinate. Thus, headings for Lidar objects are first transformed from the local to global coordinates.

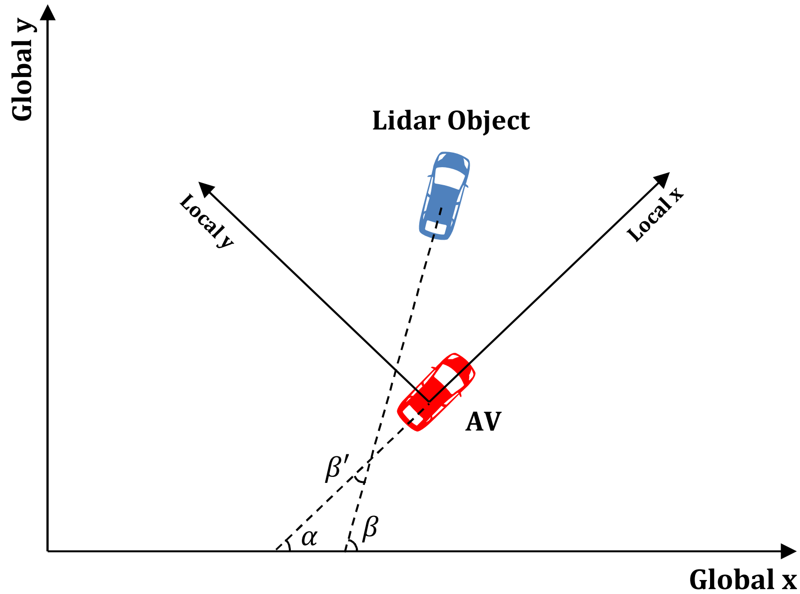


Figure 3. Heading coordinate transform for Lidar object

As illustrated in Figure 3, suppose the AV's global heading is α and the Lidar object's local heading is β′, then from basic geometry we know that the global heading of the Lidar object is given as in Equation (2):

$$\beta = \alpha + \beta' \tag{2}$$

Trajectory view videos combined with camera videos enable us to intuitively understand the driving environment and driving behavior. As an example, a snapshot of the trajectory view video for Segment 436 is shown in Figure 4. All 4 types of objects (AV vehicle, HV vehicle, cyclists and pedestrians) can be clearly identified in this figure. The box sizes for all objects are proportional. Note that each object's global unique ID has been substituted with a local ID starting from 0 (AV's local ID is always 0).

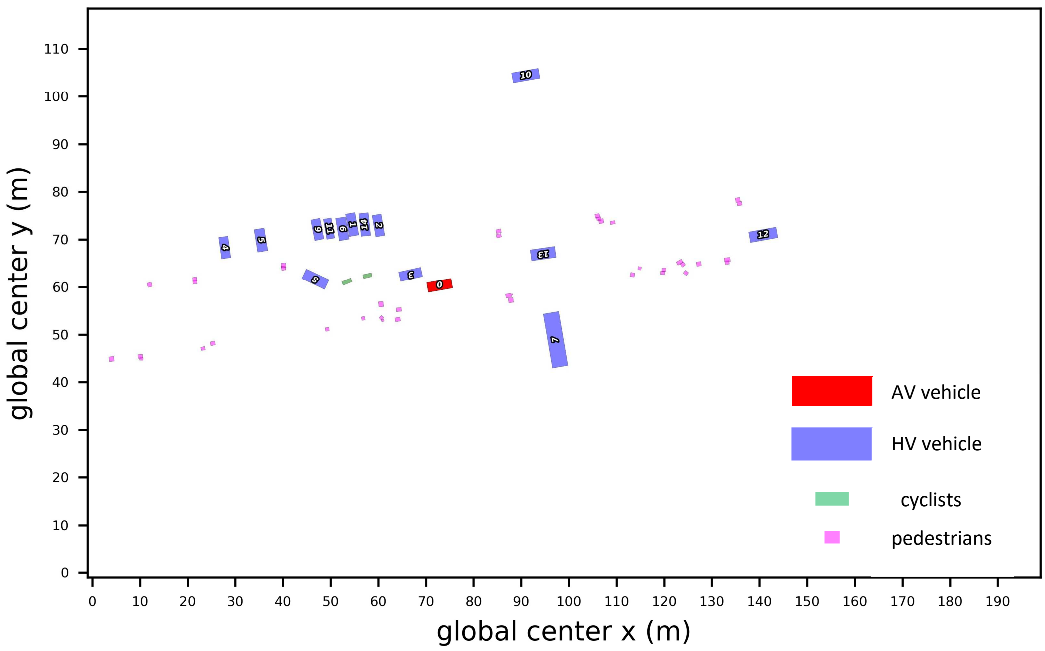


Figure 4. Example: A trajectory view video screenshot of Segment 436

One question important for the Lidar information in the Waymo Open Dataset is: what is the detection range of Lidar for different types of objects in different driving environment? Since Waymo might have truncated the data range here we only focus on the available detection range in the provided dataset. To answer this question, the maximum distance of each object to the AV vehicle is computed for each segment, and the resulting statistics are shown in Table 3. Note that the median value is used instead of the mean value because the median is generally more robust towards outliers. Table 3 shows that the Lidar's detection range for vehicles is relatively stable with a small mean absolute deviation of 1.21m, and that the Lidar's median detection range for all three objects are rather similar, i.e., 77 ~ 80 m.

Table 3. Lidar detection range statistics for the 1000 segments

Statistic	Vehicle	Cyclist	Pedestrian
Max (m)	86.61	78.13	79.39
Min (m)	61.08	7.91	18.61
Median (m)	79.92	77.38	77.43
Mean absolute deviation (m)	1.21	11.29	7.31

4.3 CF vehicle pair selection

4.3.1 Selecting car following vehicle pairs based on videos

One might think this step is unnecessary because it seems much faster and more accurate to extract CF pairs using some automatic detection method. However, according to our observation, even if the accurate follower-leader pair is detected, the follower might not be in CF state due to disruptions like lane changing, turning, parking, traffic light, road signs, etc. Without excluding these factors, some of the extracted trajectories will be disrupted by exogenous factors, and thus not suitable for research related to car following behavior and modelling. To ensure the quality and reliability, we have manually extracted CF pairs, despite the fact that this manual method is quite labor intensive and time consuming.

For different research purposes, the CF pair extraction methods should be designed accordingly. As an example, here we focus on the driving behavior of light duty vehicles under constrained conditions. More specifically, 6 rules are developed and implemented in filtering out unsuitable vehicle pairs, as summarized in Table 4. For each rule, one or more vehicle pair examples are presented. Each example is represented by the segment ID and IDs of both the follower and leader vehicle. The reader is referred to the trajectory view videos or camera videos for better understanding of these excluded vehicle pairs, some of which can be quite misleading.

Table 4. CF pairs excluding rules (a smaller index has a higher priority) and examples

Rule ID	Rule descriptions	Examples (not exhaustive)
1	Exclude if there is no leader or follower	Segment1-Vehicle 0
2	Exclude if the follower or leader is off the Lidar detection range (disappear from the video) for some time	Segment55-Follower3-Leader0
3	Exclude if the leader or follower is a bus or heavy truck	Segment47-Follower144-Leader0
4	Exclude if the follower changes its leader (either the follower or the leader changes its lane)	Segment15-Follower0-Leader35, Segment391-Follower0-Leader5
5	Exclude if follower remains standstill during the entire segment	Segment81-Follower48-Leader0
6	Exclude if the car following state is interrupted by turning, parking, stop signs, traffic signals, pedestrians, or other obstacles	Segment104 Follower0 Leader9, Segment436 Follower0 Leader12, Segment61 Follower0 Leader2, Segment185 Follower15 Leader0, Segment185 Follower15 Leader0, Segment673-Followre0-Leader5

For Rule 6, occasionally it is hard to distinguish between a proper CF pair and an abnormal CF pair by watching videos. However, according to the car following theory, if a vehicle is in the CF state (as a follower), generally its velocity will increase as spacing increases and remain around the desired speed when the spacing is sufficiently large. Thus, the velocity-spacing plot can be used to help us scrutinize the relationship of their trajectories, and decide whether a pair of trajectories is significantly disrupted by those exogenous factors listed in Rule 6. Several typical examples are shown in Figure 5.

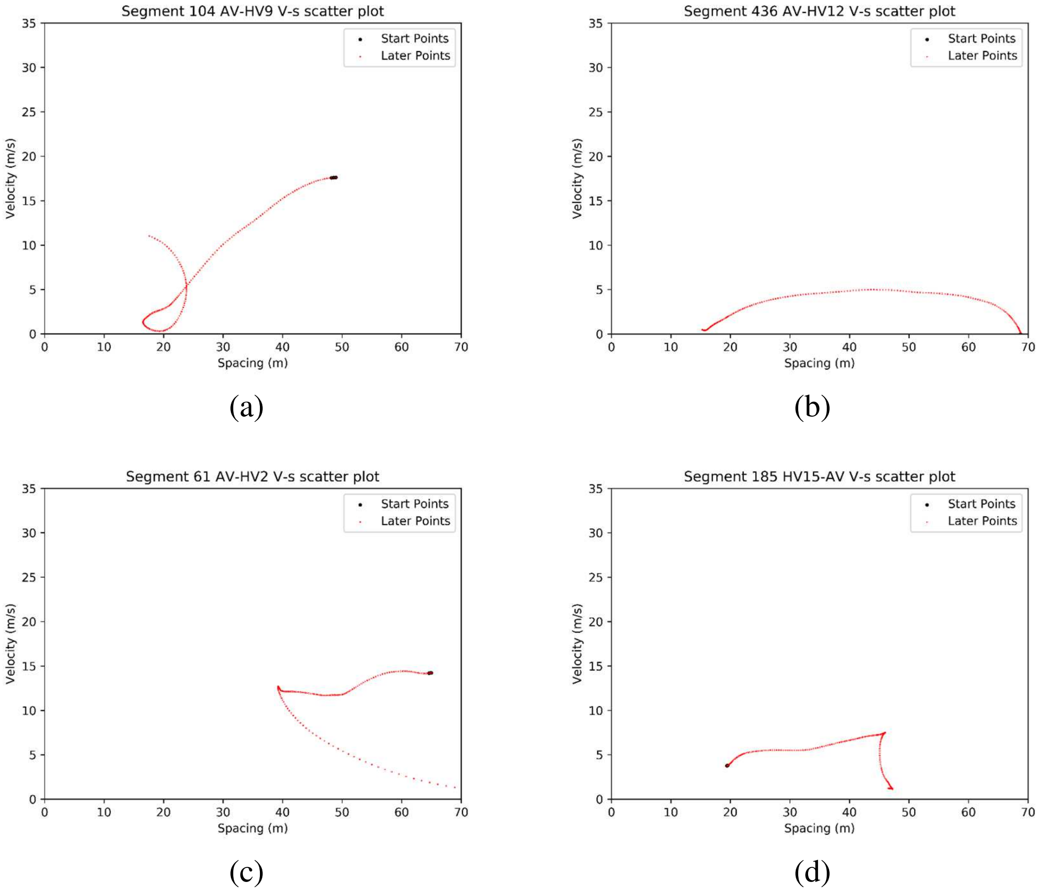


Figure 5 Examples of CF verification using the velocity-spacing plot

Among 1000 segments, we observe three types of vehicle pair: AV-HV (196) where AV is following an HV, HV-AV (274) where HV is following an AV, and HV-HV (1032) where both vehicles are HV. The number in parentheses is the number of appropriate CF pairs for each group. Since CF pairs where AV is involved are the most valuable part in this dataset, it is important to keep the sample size that contains AV as large as possible. Thus, we were very cautious in excluding any AV-HV or HV-AV CF pair and only did so when we were able to explicitly give the reason why it is not suitable for CF behavior research. The number of vehicle pairs excluded corresponding to each rule is presented in Figure 6. However, for HV-HV group, normally there are multiple HV-HV CF pairs in one segment, which leads to a much larger sample size for this type of CF pairs. To avoid the dominance of HV-HV CF pairs in the final dataset, when selecting HV-HV group we only kept the CF pairs with high quality while discarding many cases with questionable quality. Therefore, providing reasons for each of those excluded HV-HV pair would be labor-intensive and with little value.

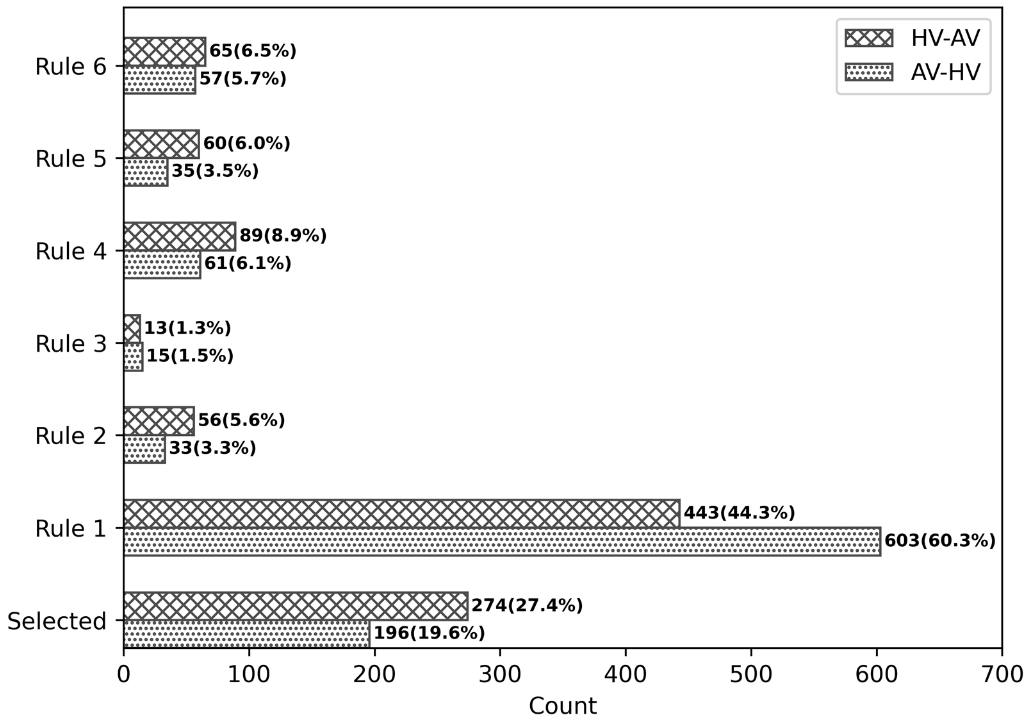


Figure 6. Number of CF pairs selected and CF pairs excluded for each rule

In summary, a total of 2228 vehicles' trajectories including 432971 data points are extracted. They form 1502 CF pairs, amongst which 470 pairs are AV-related pairs. The trajectories of these 1502 CF pairs are suitable for examining CF behavior in the AV related environment.

4.3.2 1-d longitudinal trajectory

With a focus on CF behavior analysis, it is necessary to convert the data format from the original global position to the one-dimension longitudinal coordinate. For each CF vehicle pair, this is achieved by following the steps below:

Step 1: calculate the cumulative displacement for the follower and the leader;
Step 2: calculate the initial distance between the follower and the leader;
Step 3: assign the starting position (the origin) of the follower as 0;
Step 4: calculate the position of the follower and the leader at each time step with respect to the origin.

Note that the original dataset provides both the global position (x-y-z coordinates) and the corresponding speed, where the speed is derived from the position and then processed (how the derived speed is processed is unknown) by Waymo. To retain as much information as possible, we keep both the position data (called the position-based data hereafter) and the speed data (called the speed-based data hereafter) during the data processing. Thus, two sets of position-speed-acceleration exist in the processed data: a) one set is the position-based data in which speed and acceleration are computed from position by differentiation; b) one set is the speed-based data in which position (by integration) and acceleration (by differentiation) are derived from speed.

Specifically, in Step 1, the cumulative displacement for the position-based data is computed using cumulative sum of the individual displacement, while the cumulative displacement for the speed-based data is computed by integrating individual speed, as shown in Equation (3) and Equation (4), respectively. The remaining steps are the same for both data sources.

$$X_p^i = \sum_{k=1}^{i-1} \sqrt{(x_g^{k+1} - x_g^k)^2 + (y_g^{k+1} - y_g^k)^2 + (z_g^{k+1} - z_g^k)^2} \quad (3)$$

where X_p^i is the position-based cumulative displacement of the ith point, while x_g^k, y_g^k, z_g^k are the vehicle's global x/y/z positions of the kth point, respectively.

$$X_s^i = \sum_{k=1}^{i-1} \frac{v_{k+1}+v_k}{2} * 0.1 \quad (4)$$

where X_s^i is the speed-based cumulative displacement of the ith point, while v_k is the vehicle's speed of the kth point and 0.1 is the time resolution.

5. Trajectory data quality assessment

5.1 Consistency analysis

Trajectory consistency is an important index in evaluating the quality of trajectory data. It includes internal consistency and platoon consistency: the internal consistency is whether or not the differentiation of positions yields consistent speeds and accelerations, and the platoon consistency verifies whether the inter-vehicle spacing drawn by the trajectories estimated for a pair of vehicles is consistent with the actual one(Punzo et al., 2005, Punzo et al., 2011). Since in the Waymo Open Dataset, the trajectories and actual spacings are both calculated from global positions, the platoon consistency issue does not exist. Thus, only the internal consistency is analyzed in this paper.

As an example, Figure 7 shows a position/speed/acceleration profile of both the position-based and the speed-based data. While the trajectories from both the data sources match each other quite well, small deviations in the speed profile and large deviations in the acceleration profile can be observed. This example illustrates the necessity of verifying the internal consistency of the vehicle trajectories extracted from the Waymo Open Dataset.

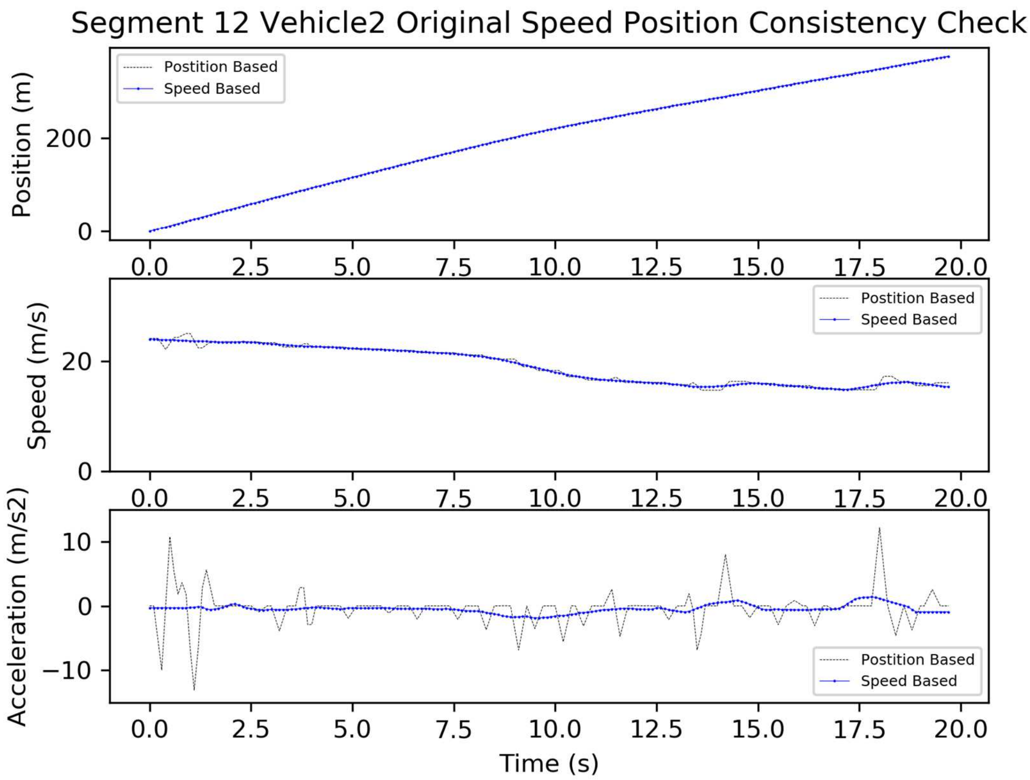


Figure 7. The consistency between the position-based and the speed-based data

Quantitative consistency analysis in terms of position, speed and acceleration is then conducted respectively, using the consistency index, which is defined as Root Mean Squared Error (RMSE) as shown in Equation (5):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Z_s^i - Z_p^i)^2} \tag{5}$$

where N is the number of observations in the vehicle's trajectory, Z_s^i is a speed-based measurement (position, speed, or acceleration) of the ith point, while Z_p^i is the corresponding position-based measurement. In Table 5, four statistics, i.e. maximum, minimum, mean, and standard deviation of each RMSE are presented. Although the RMSE for position consistency is relatively small, the average RMSE for speed is noticeable and the average RMSE for acceleration consistency is as large as $0.9m/s^2$. In light of the large magnitude of RMSE, it is concluded that position-based data and speed-based data are inconsistent and considerable data process work had been done by Waymo to derive the speed from the position (again, how exactly the derived speed is processed is unknown). Therefore, the speed-data provided by Waymo are generally not recommended to use (more reasons for this recommendation are given in Section 6.1 and Section 6.2).

Table 5. Results of the internal consistency analysis

Statistic	Position RMSE (m)	Speed RMSE (m/s)	Acceleration RMSE (m/s^2)
Max	0.72	1.57	11.43
Min	0.00	0.00	0.00

Mean	0.08	0.14	0.90
Std	0.06	0.11	0.77

5.2 Jerk value analysis

To acquire a better understanding on the data quality of the Waymo Open Dataset, we also perform jerk value analysis. Following the work by Punzo et al. (2011), extreme jerk values and sign variation are analyzed. The absolute Jerk values larger than $15m/s^3$ are considered as not physically feasible. Also, more than one sign inversion in a one-second window is defined as anomalous jerk sign inversion. Table 6 shows the statistic results for both the position-based data and the speed-based data. In the position-based data the proportion of anomaly jerk values is as large as 5.3%. However, in the speed-based data, the proportion of anomaly jerk is drastically smaller, i.e., 0.00439%, and similarly, the proportion of anomaly jerk sign inversion in the speed-based data is also considerably reduced. Additionally, the extreme jerk values (i.e., the maximum, and the minimum) contained in the speed-based data are also significantly smaller than those contained in the position-based data, which indicates that the speed data provided by Waymo has already been reasonably processed. However, it is clear that there still exists unreasonable jerk values in the speed-based data.

Table 6. Jerk analysis results for the position-based data and the speed-based data

Jerk analysis index	Position based	Speed based
Anomaly jerk proportion (%)	5.3	0.00439
Maximum jerk (m/s^3)	> 100	30.71
Minimum jerk (m/s^3)	< -100	-19.81
Anomaly jerk sign inversion proportion (%)	86.1	37.2

Table 7. NGSIM data internal consistency and jerk analysis results

Data location	Position consistency RMSE (m)	Speed consistency RMSE (m/s)	Anomaly jerk proportion (%)	Anomaly jerk sign inversion proportion (%)
I-80 Freeway	0.39	1.55	12.9	87.3
US-101 Freeway	0.10	0.86	8.4	87.0
Lankershim Arterial	6.81	2.53	16.0	93.8
Peachtree Arterial	4.30	1.25	9.1	79.7

Note: these values are averaged values across all lanes for each location (Punzo et al., 2011)

It is interesting to compare the results of consistency analysis and jerk value analysis between the NGSIM dataset (as presented in Table 7) and the Waymo Open Dataset. Clearly, consistency RMSE values and anomaly jerk pattern proportion for the Waymo Open Dataset are smaller (even using the position-based data) than those for the NGSIM dataset. Thus, from this perspective we can conclude that the quality of the Waymo Open Dataset is better than that of the NGSIM dataset.

5.3 Trajectory completeness assessment

Another important aspect of the trajectory quality assessment is trajectory completeness, i.e. whether the trajectory contains sufficient number of driving regimes for car following model development and calibration. A trajectory is complete if all 6 driving regimes are included: Cruising at desired speed (C), free acceleration (Fa), following the leader at a constant speed (F), accelerating behind a leader (A), decelerating behind a leader (D), and standing behind a leader (S) (Treiber and Kesting, 2013a, Sharma et al., 2018). Sharma et al. (2018) proposed a pattern recognition algorithm for assessing trajectory completeness, and the same algorithm is applied for the Waymo Open Dataset. Result is shown in Figure 8. Note that the focus of our analysis is limited to classifying driving regimes contained in the trajectories. For further understanding the impact of trajectory completeness on CF model calibration, readers are referred to (Sharma et al., 2019a).

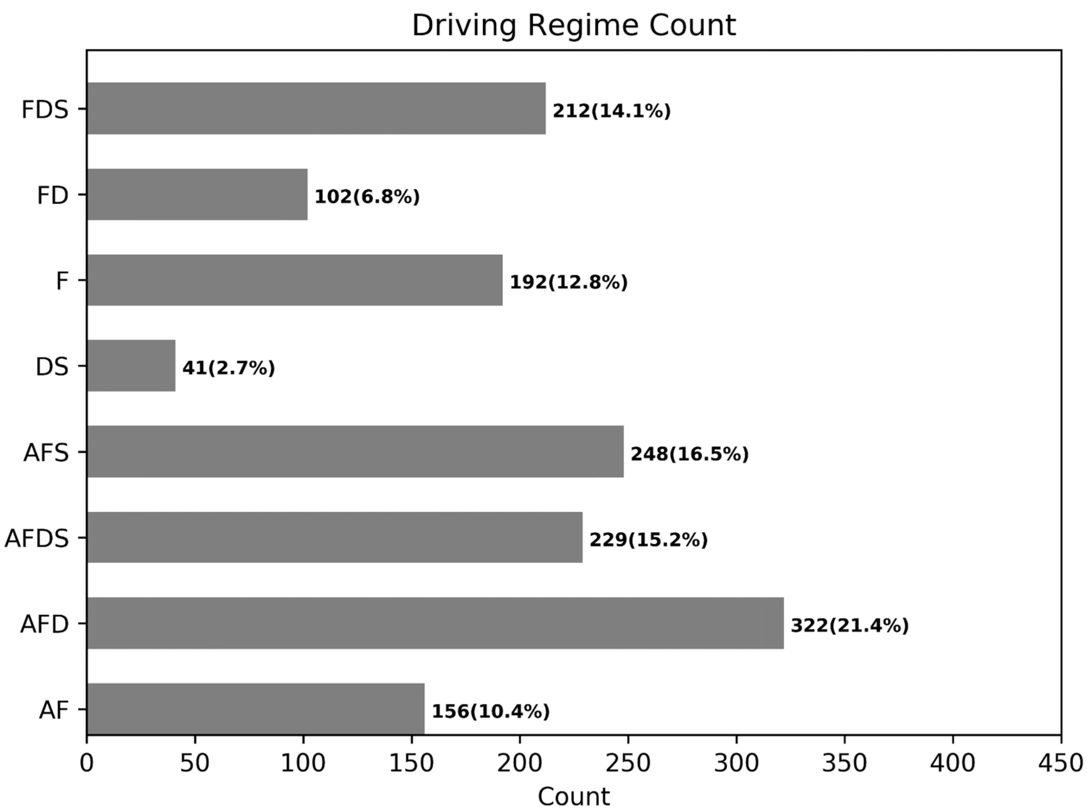


Figure 8 Driving regime classification results

In Figure 8, the number (and percentage) of each driving regime appearing in the extracted Waymo trajectories is presented. Basically, no trajectory is complete, which is the same as in NGSIM data (Sharma et al., 2018). Also, in both Waymo data and NGSIM, trajectories containing both acceleration and deceleration (AFD or AFDS) are most frequent. However, two obvious differences in terms of trajectory completeness between Waymo data and NGSIM are: 1) the lowest level of completeness in NGSIM data is AFD, while in Waymo data the lowest level is F due to the short length of each segment; and 2)77% of the NGSIM data lacks the standstill regime, while in Waymo data only 51.4% of the trajectories do not have the standstill regime, which is due to the fact that in Waymo data many cases are in signalized urban streets, while NGSIM data were collected from freeways.

These driving regime classifications can be valuable for further research on driving behavior of both AV and HV vehicles, and CF model develop in particular. Thus, such classification information is included in the final Waymo dataset processed by us.

6. Trajectory quality enhancing

Despite that the Waymo dataset has shown a better quality than the NGSIM data, how Waymo processed the position-based data to obtain the speed-based data is still unknown; more importantly, we have demonstrated that there exists significant inconsistency in speed and acceleration between the position-based data and the speed-based data. Moreover, as shown previously, in the speed-based data, there still exists noticeable anomaly data (37.2% anomaly jerk sign inversion). Therefore, the speed-based data are not recommended to use, and the position-based data should be generally preferred after the position-based data's quality is enhanced. In this section we propose methods to remove outliers and filter noise contained in the position-based data.

In the literature, the methods for enhancing trajectory data's quality can be classified into two types: the one-step methods and the multistep methods. The one-step methods include simple/exponential/kernel-based moving average (Duret et al., 2008, Hamdar and Mahmassani, 2008, Ossen and Hoogendoorn, 2008, Thiemann et al., 2008), local function fit method such as locally weighted regression (Toledo et al., 2007) and spline smoothing method (Vieira da Rocha et al., 2015). The main drawback of one-step methods is that outlier or noise is not locally identified and normal data points are excessively smoothed (Rafati Fard et al., 2017). This issue is solved in the multistep methods (Montanino and Punzo, 2013, Montanino and Punzo, 2015, Rafati Fard et al., 2017) where outlier or noise is located and dealt with separately. The multistep methods are adopted in this research. More specifically, an optimization-based outlier removal method is first proposed, and then the data are denoised by a wavelet filter method.

6.1 Outlier removal

The first step of the proposed data quality enhancement method is outlier removal. This step is important because a) outliers cannot be removed by denoising; and b) the data will be significantly biased due to the existence of outliers. Traditionally for outlier removal, each time only a single value is removed as an outlier and replaced by a new value (e.g., via interpolation). This practice is problematic because of its ineffectiveness of removing outliers: outlier removal is implemented on one variable (usually speed or position), while outlier identification is done on another variable (usually acceleration). Thus, after replacing an outlier in speed or position, the new data point can still be an outlier by checking its corresponding acceleration which is calculated via differentiation. To remedy this issue, we can replace multiple values within a small window defined around the outlier instead of only replacing the outlier itself. However, the boundaries of the window can be hard to control, and may still produce new outliers.

Therefore, a simple but effective optimization-based outlier removal method is proposed in this research. Two steps are introduced: the first step is to identify the positions of outliers based on acceleration; the second step is to replace the trajectories near each outlier. At the first step, outliers are defined as points with abnormal acceleration values, where the acceleration limits, a_{min} and a_{max} are set to be -8m/s^2 and $5m/s^2$, respectively, the same as in Montanino and Punzo (2015). At the second step, the input is a sequence of trajectory points with one or multiple outliers, and the output is the same number of trajectory points without outliers. Table 8 summarizes the parameters used in the optimization model. Note that a_{min} and a_{max} are utilized again for restricting the optimized acceleration. Once an outlier is identified, trajectory points within the vicinity (defined by the window size T) of the outlier are extracted as the input to the model. While a'_{max} and a'_{min} are introduced for the formulation of objective function. They are temporary auxiliary variables in the process of searching for the optimal solution.

The main idea behind this outlier removal method is that the original trajectory with outliers should be replaced by an outlier-free trajectory as "smooth" as possible. Smoothness (strictly speaking, roughness) here is defined as the difference between the maximum acceleration and the minimum acceleration in the optimization window, which is represented by the objective function (6). Acceleration limit is represented by Constraint (7). Constraint (8) and (9) are the boundary constraints on position/speed/acceleration for the first and last point within the window, respectively. Constraint (10) and (11) are updates of position and speed, respectively. Note that when updating the position and speed, we have adopted the ballistic scheme by assuming that the acceleration is constant during each time step. Considering that the time resolution Δt is as small as 0.1 second, the constant acceleration assumption is reasonable. The ballistic scheme is widely used in the car-following modeling literature (Treiber and Kanagaraj, 2015, Osorio and Punzo, 2019, Punzo et al., 2021). Constraint (12) and (13) are imposed on auxiliary variables a'_{max} and a'_{min}. With the proposed objective function and constraints, the optimization model generates the optimal position, speed and acceleration profile by searching the feasible solution space.

$$\min(a'_{max} - a'_{min}) \tag{6}$$

$s.t.$

$$x_1 = x_1^r, v_1 = v_1^r, a_1 = a_1^r \tag{8}$$

$$x_T = x_T^r, v_T = v_T^r , a_T = a_T^r \tag{9}$$

$$x_t = x_{t-1} + 0.5 \cdot (v_t + v_{t-1}) \cdot \Delta t, t = 2, 3, \cdots, T-1 \tag{10}$$

$$v_t = v_{t-1} + a_{t-1} \cdot \Delta t, t = 2, 3, \cdots, T-1 \tag{11}$$

$$a'_{max} \geq a_t, t = 1, 2, \cdots, T \tag{12}$$

$$a'_{min} \leq a_t, t = 1, 2, \cdots, T \tag{13}$$

Table 8. Outlier removal optimization model parameters

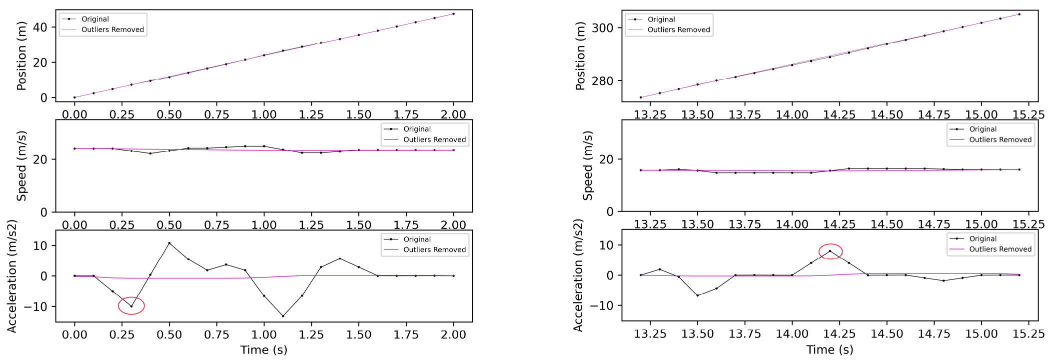
Variable	Description
a_{min}, a_{max}	Minimum/maximum acceleration limit
T	Outlier removal window size
Δt	Time step length

a_t, v_t, x_t	Optimized acceleration/speed/position at time step t
a_t^r, v_t^r, x_t^r	Actual acceleration/speed/position at time step t
a'_{min}, a'_{max}	Optimized minimum/maximum acceleration

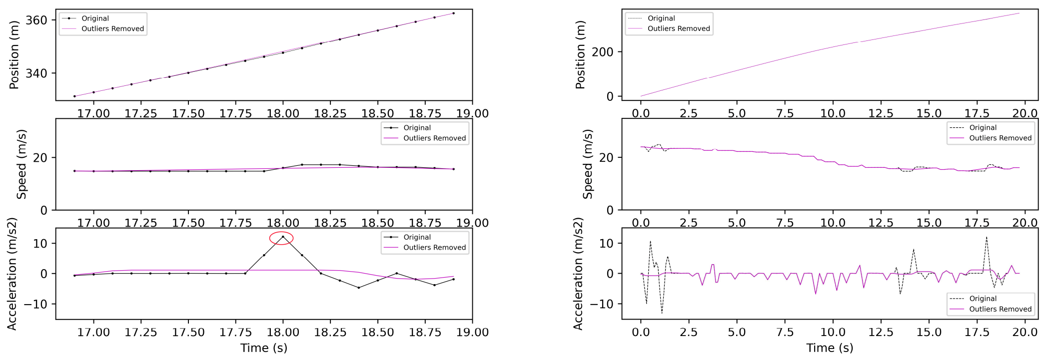
This optimization model is a linear programming problem, which can be solved efficiently using any commercial or open-source solver. The feasibility of this optimization problem depends on the data. Generally, a larger window size T will make the model more feasible. In this study, a time window of 2 seconds is sufficient for assuring the feasibility of the optimization model. Thus, T is set to be 20 because Δt is 0.1s.

The main advantage of using this optimization model for removing and replacing outliers is that a set of optimized position/speed/acceleration values are computed simultaneously, under a series of constraints, which guarantees that the resulting accelerations are within a reasonable range and that the boundary points are not new outliers.

An example of removing outliers using the optimization model is given in Figure 9. Three outliers are identified along the 20s trajectory. They appear at t = 0.3s, t = 14.2s, and t = 18.0s, respectively, and Figure 9(a-c) present the sections where each outlier locates within the 2-s window size. As shown in these figures, for each outlier the optimization-based method can reasonably generate a locally smoothed trajectory while not producing any new outlier at the boundaries. Figure 9(d) shows the overall position, speed and acceleration profiles before and after outlier removal. From this figure, we can clearly see that all the accelerations are within the normal range in the new trajectory while the speed trend is nicely preserved since the outlier removal is implemented locally.



(a) (b)



(c) (d)

Figure 9. Outlier removal for Vehicle 2 in Segment 12: (a) The outlier at 0.3s; (b) The outlier at 14.2s; (c) The outlier at 18.0s; (d) The overall result before and after the outlier removal.

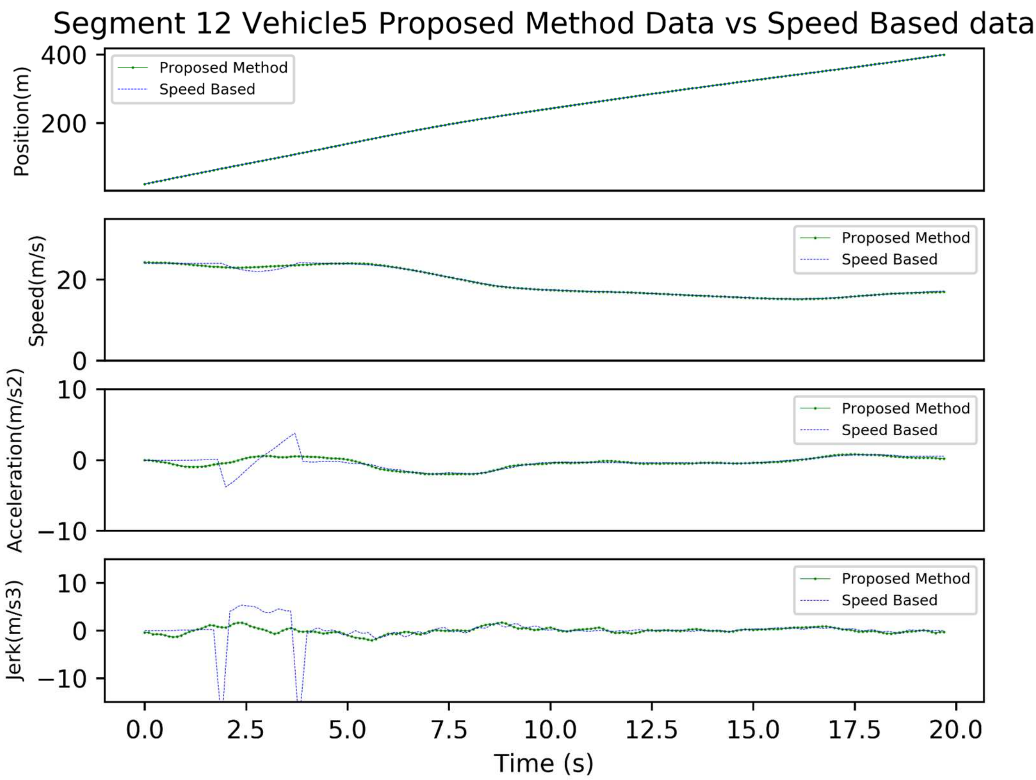


Figure 10 Trajectory of vehicle 5 in segment 12 outlier removal comparison

To compare the quality of the position-based data after removing outliers using the proposed method and that of the speed-based data, an example is given in Figure 10. In this figure, the outliers in the speed-based data can be clearly identified in the acceleration and jerk profiles, while they are completely removed in the position-based data after being processed using the proposed outlier removal method. This example together with the nice properties of the optimization-based method convincingly shows that the proposed outlier removal method is more effective and reliable than the method used by Waymo in generating the speed-based data.

6.2 Denoising

After outlier removal, although extreme values no longer exist, abnormal fluctuations in the position and speed profiles are still noticeable. For example, as observed in Figure 9(d), in the speed profile there is an apparent deceleration at around t = 9s, which lasts for about 2s; however, this deceleration trend is not consistent with the acceleration profile in the same time period. Thus, it is necessary to denoise the Waymo Open Dataset. The Wavelet denoising method is implemented here because Wavelet has been widely used in the literature as a powerful and efficient method for suppressing or eliminating noise in the data and revealing the true characteristics of the underlying signal (Coifman and Donoho, 1995, Donoho, 1995, Donoho and Johnstone, 1995, Donoho et al., 1995, Stephane, 1999, Taswell, 2000). In the literature, using wavelet transform (WT) to denoise data is called wavelet shrinkage, a concept introduced by Donoho and his collaborators (Donoho, 1995, Coifman and Donoho, 1995, Donoho and Johnstone, 1995, Donoho et al., 1995). Wavelet shrinkage can be used as a general denoising tool because its performance is unlikely to be significantly worse than that of several established non-wavelet denoising methods. Note that although WT has been frequently used in traffic flow research for detecting singularities in the data (e.g., stop-and-go oscillations(Zheng et al., 2011b, Zheng et al., 2011a), lane changing(Zheng and Washington, 2012, Ali et al., 2020b), driver's response time(Sharma et al., 2019c), this is one of the first studies that use WT to denoise vehicular trajectory data (Rafati Fard et al., 2017). Compared to the optimization-based filtering method by Montanino and Punzo (2015), wavelet filtering method has fewer parameters (only needs to choose the wavelet and threshold) and the results are more controllable. Moreover, since the existed wavelet filtering methods are designed for NGSIM dataset, they did not perform well enough for the Waymo dataset according to our experiments. Therefore, it is more appropriate to develop a wavelet filtering method that is adequately suitable for denoising the Waymo Open Dataset. Note that it is totally possible that there are methods that can give a better denoising performance than the designed method. However, comparing different filtering methods' performances can be tricky due to the simple fact that we often do not have the luxury of knowing the ground truth.

The basic idea of wavelet shrinkage is intuitive: WT decomposes the signal into two components at various scales: the high frequency component (contained in the detail coefficients) and the low frequency component (contained in the approximation coefficients), and it is natural to do some modification to the detail coefficients to remove or suppress the noise before we reconstruct the signal. When using WT to denoise a signal, three basic steps are involved: (1) Choose a Wavelet (e.g., Harr, Symlet, Daubechies, Coiflet, etc.) and use it to decompose (via WT) the signal into the approximation part and the detail part at different scales; (2) Apply coefficients thresholding to the detail coefficients using some shrinkage methods (e.g., naïve thresholding, hard thresholding and soft thresholding and SURE thresholding (Nason, 2008)); and (3) Reconstruct the signal based on altered coefficients(via inverse WT).

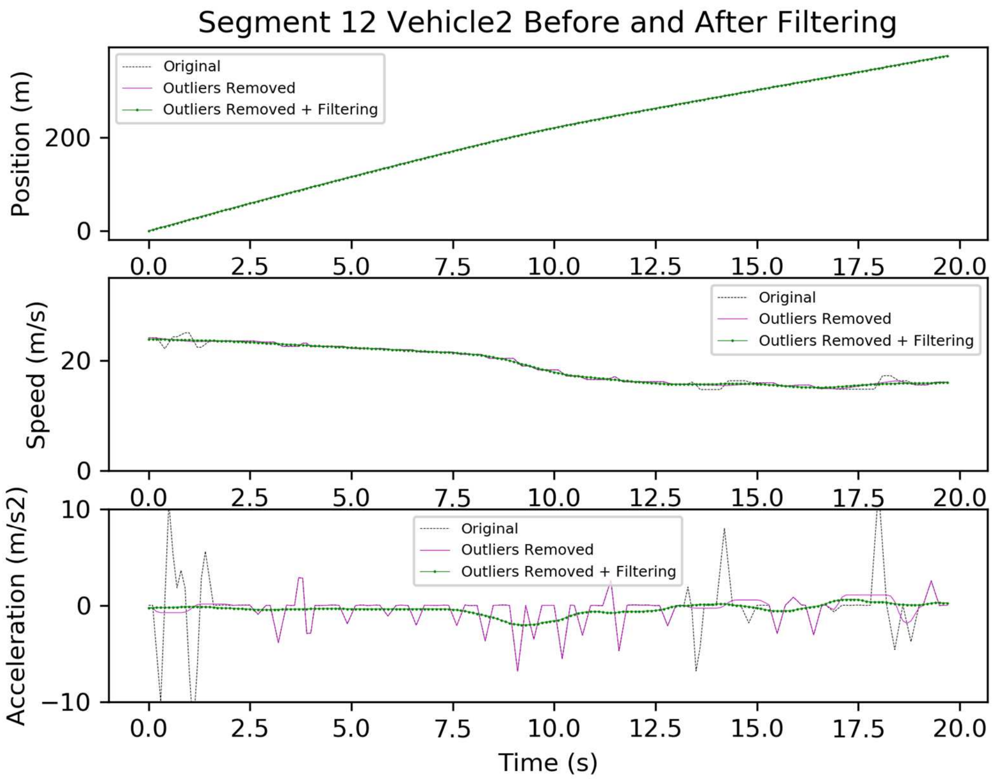


Figure 11. Trajectory of vehicle 2 in segment 12 before and after filtering

In this research, daubechies6 Wavelet is selected after some trial and error. The maximum decomposition level is set to the maximum (4 in our cases). Naïve thresholding (setting all the detail coefficients to 0) is adopted as the shrinkage method because the duration of the signal is short (each segment is just about 20s) and the time resolution is also relatively high (0.1s) and within this short duration there are no significant structural changes in the signal (i.e., the speed profile). Consequentially, detail coefficients at each scale are quite small. By using PyWavelets python package (Lee et al., 2019), the wavelet filtering method is implemented on speed profile, which is the mean speed between two consecutive time steps derived via differentiation from the position-based data. For the convenience of comparison, the result of the same vehicle from outlier removal is shown in Figure 11. As seen from this figure, the abnormal humps on the speed profile disappear and the deceleration trend is consistent in the acceleration profile. It is also noteworthy that the applied wavelet filtering method does not alter the total travel distance (on average the difference is as small as 0.0483%). Together with the proposed outlier removal method and the wavelet denoising we have processed the position data of the Waymo Open Dataset.

To further demonstrate the effectiveness of the proposed outlier removal method and wavelet denoising method, we next quantitatively compare the processed data and the speed-based data since the ground truth is unknown. More specifically, we aim to answer two questions here: (a) is the degree of denoising of the proposed method more than that of Waymo's speed-based data? In other words, does the proposed method over-smooth the data? and (b) does the proposed method lead to more reasonable jerk values? To answer the first question, the RMSE-based consistency index as defined in Equation (5) is used to measure the difference between the processed data and the position-based data (see Table 9), and we then compare the result with the consistency analysis between the speed-based data and the position-based data (see Table 5). The position RMSE of the processed data (0.05) is less than that of the speed-based data (0.08), while the speed RMSE and acceleration RMSE are slightly larger. Overall, the degree of denoising between the processed data and the speed-based data is similar.

Table 9 Consistency index between the processed data and the position-based data

Statistic	Position RMSE (m)	Speed RMSE (m/s)	Acceleration RMSE (m/s^2)
Max	0.62	1.63	11.42
Min	0.00	0.00	0.00
Mean	0.05	0.15	0.92
Std	0.04	0.13	0.77

For the second question, the same jerk analysis is conducted on the processed data, and the result is presented in Table 10. In the processed data, no anomaly jerk is found, as can be verified by the maximum jerk (9.32 m/s^3) and the minimum jerk (-9.94 m/s^3). Additionally, anomaly jerk sign inversion proportion has nearly dropped in half from 37.2% (for the speed-based data, see Table 6) to 20.4%. Therefore, we can conclude that the processed data is more reasonable than the speed-based data in terms of jerk values.

Table 10 Jerk analysis results for the processed data

Index	The processed data
Anomaly jerk proportion (%)	0.00
Maximum jerk (m/s^3)	9.32
Minimum jerk (m/s^3)	-9.94
Anomaly jerk sign inversion proportion (%)	20.4

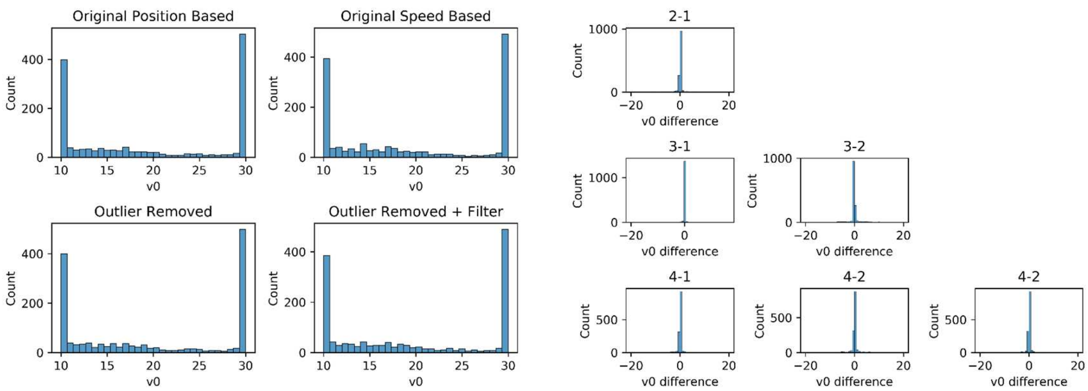
6.3 Effect of outlier removal and denoising

In this section, we demonstrate the effect of removing outliers and filtering noise from the Waymo data on car-following model calibration. More specifically, IDM (shown in Equation (14)) is selected because of its popularity in the recent literature.

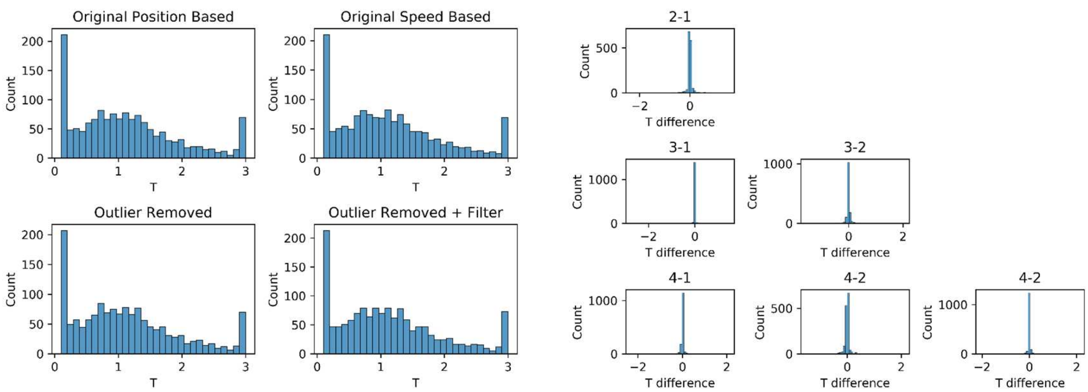
$$a_n(t) = \boldsymbol{a_n} \cdot \left[1 - \left(\frac{v_n(t)}{\boldsymbol{V_{0,n}}}\right)^{\boldsymbol{\delta_n}} - \left(\frac{\boldsymbol{s_{0,n}} + \boldsymbol{T_n} \cdot v_n(t) - \frac{v_n(t) \cdot \Delta v_n(t)}{2 \cdot \sqrt{\boldsymbol{a_n} \cdot \boldsymbol{b_n}}}}{\Delta x_n(t) - l_{n-1}}\right)^2\right] \quad (14)$$

where $v_n(t)$ and $a_n(t)$ are a follower vehicle (n)'s speed and acceleration at time *t*, Δx_n and $\Delta v_n(t)$ are the inter-vehicle spacing and speed difference from the leader vehicle, and l_{n-1} is the leader vehicle length. Model parameters are in bold, including desired speed of the vehicle (V_0; unit: m/s), free acceleration exponent (δ), desired time gap (T; $unit$: s), minimum gap (s_0; unit: m), maximum acceleration (a; $unit$: m/s^2), and desired deceleration of vehicle (b; $unit$: m/s^2). δ is set as 4 in this study as recommended in the literature (Treiber and Kesting, 2013b).

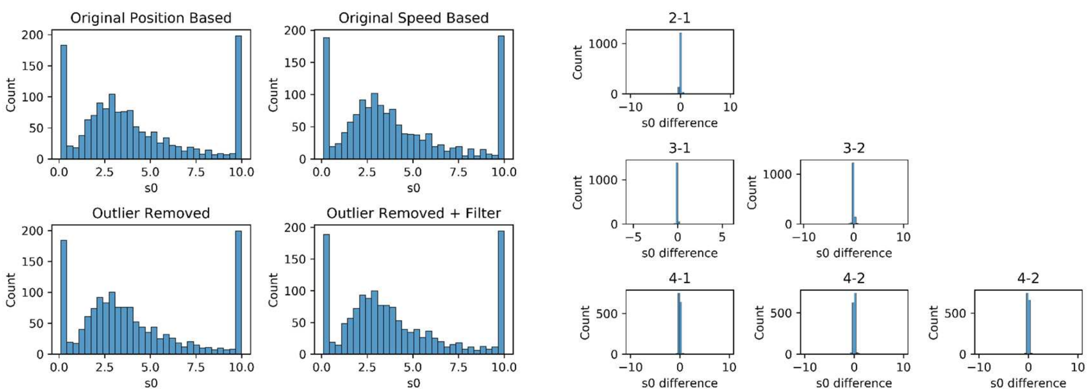
Four versions of the Waymo dataset are considered: the position-based data (Group 1), the speed-based data (Group 2), the position-based data with outliers being removed (Group 3), and the position-based data with outliers being removed plus noise being filtered (Group 4), and each group contains 1502 pairs of vehicle trajectories. IDM is calibrated separately using every paired trajectory in each group. In the calibration setting, the global approach is used where each objective function evaluation is a simulation run (Ciuffo et al., 2008). In the objective function, spacing is chosen as the measure of performance and RMSE as the goodness-of-fit, as recommended by . The calibration range of IDM parameters are set as follows: desired speed v_0 [10, 30], desired time gap T[0.1, 3], minimum gap s_0 [0.1, 10], maximum acceleration a [0.5, 5], desired deceleration b [0.5, 5]. The distributions of the parameter values calibrated for each data source are shown in Figure 12. Meanwhile, the pair-wise group differences in the calibrated parameter values are also presented in this figure.



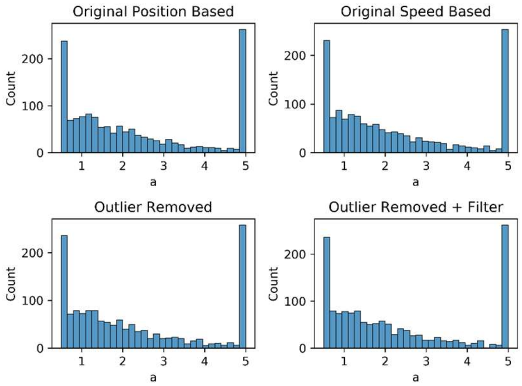
(a) v_0 histogram (b) v_0 difference histogram



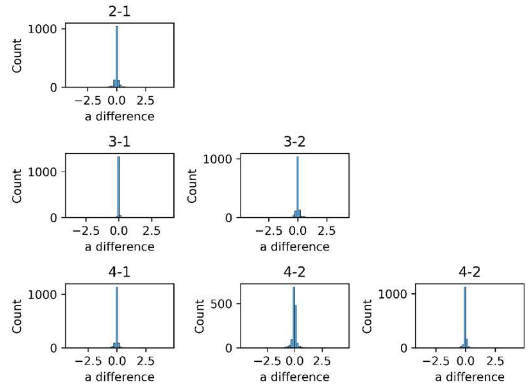
(c) T histogram (d) T difference histogram



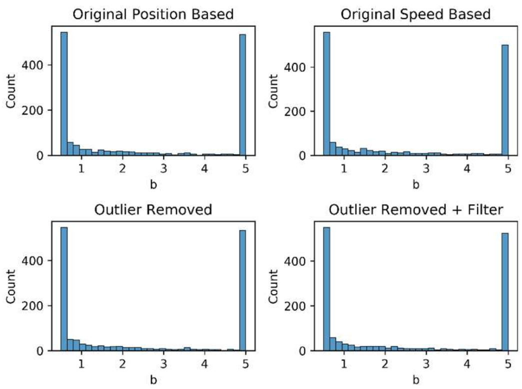
(e) s_0 histogram (f) s_0 difference histogram



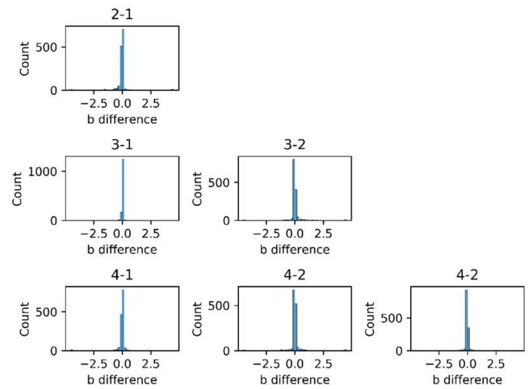
(g) a histogram



(h) a difference histogram



(i) b histogram



(j) b difference histogram

Figure 12 The distributions of the parameter values calibrated for each data source and the pair-wise group differences

The objective here is to detect if there are any significant differences between the calibrated parameters from different groups of data. Note that both the visual inspection and tests like the Kolmogorov-Smirnov (KS) test are misleading in our case. With either KS or visual inspection, all the samples are mixed together and treated as a whole. However, by doing so, a key feature for the sample in our study would be totally discarded, that is, in our case it is actually a "before-after" comparison for each pair of vehicles. In such a situation, paired statistical tests should be used because they give us more accurate and more reliable results. Therefore, instead of unpaired tests like KS test or unpaired t test, paired tests like Wilcoxon test or paired t test are more suitable in our analysis. In addition, since the distributions of our sample are clearly different from normal distributions (using Shapiro-Wilk test), non-parametric test like Wilcoxon test is more reliable. Thus, in our study, Wilcoxon test is employed. The test result is summarized in Table 11.

Table 11 The Wilcoxon test p-values for each group pair (the numbers in bold are p values less than 0.05)

Pair	v_0	T	s_0	a	b
Group 2 - Group 1	0.143	**< 0.001**	0.909	**0.019**	**< 0.001**
Group 3 - Group 1	0.613	**0.005**	0.228	**0.020**	0.771

Group 3 - Group 2	0.383	**< 0.001**	0.740	0.130	**< 0.001**
Group 4 - Group 1	0.861	**0.004**	0.441	**< 0.001**	**0.012**
Group 4 - Group 2	0.095	**< 0.001**	0.368	**0.022**	**< 0.001**
Group 4 - Group 3	0.447	**0.011**	0.168	**< 0.001**	0.089

As shown in Table 11, about half of the paired group comparisons are statistically significant at a 95% confidence level, which indicates that outlier and noise in the Waymo data can indeed influence car following model calibration results. Moreover, it is interesting to note that the differences in v_0 and s_0 between each group pair are always not significant while the differences in T and a are almost always strongly significant. This observation implies that the impact of data outliers and noise on v_0 and s_0 is negligible while T and a are sensitive towards outliers and noise in the trajectory data. This finding is consistent with conclusions given by Punzo et al. (2015). Punzo et al. (2015) investigated the relative importance of IDM parameters by analyzing the contribution of each parameter to the objective function, and concluded that desired time gap T contributes most to the variance of RMSE, followed by maximum acceleration a if the free acceleration exponent (δ) in IDM is not considered.

Overall, our analysis clearly shows that when using the Waymo data in modelling car following dynamics, outliers and noise in the data should be carefully removed and filtered.

7. Conclusions

This research has processed, assessed and further enhanced a representative of the AV-oriented empirical datasets, the Waymo Open Dataset, for driving behavior research with a focus on car following dynamics. The original dataset is re-structured and transformed to a user-friendly tabular format trajectory data with 25 essential attributes. Camera videos and trajectory view animations are generated for qualitative verification. Car following pairs are carefully selected for three groups (196 pairs for AV-HV, 274 pairs for HV-AV, 1032 pairs for HV-HV) respectively to avoid disruption caused by exogenous factors. Consistency analysis shows that the dataset itself is not internally consistent, and jerk analysis reveals that a large proportion of anomalies exist in the position-based data and a smaller but still significant portion exists in the speed-based data. Moreover, our trajectory completeness analysis suggests that the trajectories in the Waymo Open Dataset are all incomplete. Driving regimes contained in each trajectory are explicitly identified and included in the processed dataset for the convenience of future research on car following dynamics.

The trajectory data are further enhanced by using an optimization-based outlier removal method and a wavelet denoising method. The linear programming optimization model in the outlier removal method can be implemented efficiently and guarantee that the resulted trajectory is outlier-free. A wavelet denoising method is applied on the data to filter out noise. By comparing with Waymo's speed-based data, our denoised data have similar consistency index but with fewer anomaly jerk values. Additionally, we have tested the impact of data outliers and noise on IDM calibration, and revealed significant differences in parameter values for desired time gap T and maximum acceleration a.

Overall, our processed and enhanced Waymo Open Dataset contains all important information related to driving behavior of AV and surrounding vehicles and other road users. Such information has been integrated into a single and user-friendly file, which is easy for traffic flow researchers to use. Moreover, our processed Waymo Open Dataset has a higher data quality than the original dataset because the outliers have been removed, noise has been filtered, its consistency has been checked, and the trajectory completeness has been analyzed. We believe, this easy-to-use, high-quality, and information-rich dataset for mixed traffic can potentially play an important role in AV-oriented traffic flow research similar to that of NGSIM in HV-focused traffic flow research, and catalyze research progress on AV's impact in mixed traffic. Note that when using the processed Waymo Open Dataset, if small changes in acceleration are important for the research question of interest (e.g., vehicle fuel consumption, emissions, etc.), we recommend using the processed data without the denoising step because our wavelet denoising method can be regarded as too aggressive for these types of research questions. Instead, researchers can consider using a more conservative wavelet denoising method (e.g., soft thresholding, hard thresholding, SURE, etc.). To facilitate this, in our final dataset, we provide both the version with outlier removal and wavelet denoising, and the version with outlier removal but without wavelet denoising.

Besides sharing the dataset used in the analysis for this paper, which is primarily to be used in studies focusing on CF behavior of light-duty vehicles, we have also published another version (Version 2) of the processed dataset, which also contains the trajectories of 111 CF pairs where large vehicles are involved (related to Rule 3). Regarding rule 4 where the leader changes, we have carefully checked the dataset and found that the sample size (4 pairs for AV and 11 pairs for HV) is too small to support studies related to lane changing behavior. Therefore, they are not included in the second version of the processed dataset.

Similarly, to better support other researchers in extracting their own trajectories, the data processing codes for this paper have been shared in Version 2 of the published dataset (https://data.mendeley.com/datasets/wfn2c3437n/2). The shared codes include those for data-restructuring, camera video visualization, top view video visualization, CF pair selection. Moreover, the codes also incorporate the developed outlier removing and wavelet denoising method. Thus, other researchers can easily reproduce the results of this research and potentially use the methods on other trajectory datasets.

Acknowledgements

The authors would like to thank the Waymo team for sharing the data and for answering our questions. Xiangwang Hu's involvement in this research was partially supported by the China Scholarship Council (CSC), Zuduo Zheng's involvement was partially funded by the Australian Research Council (ARC) through the Discover Project (DP210102970); Danjue Chen's involvement by NSF CMMI Award # 118286, and Jian Sun's involvement by the National Natural Science Foundation of China (U1764261).

Appendix A: output table attributes description

Attribute	Description
'segment_id'	Integer number, from 1 to 1000
'frame_label'	Integer number, from 1 to around 200
'time_of_day'	String, 'Day'/'Dawn'/'Dusk'/'Night'
'location'	String, abbreviated names of US cities
'weather'	String, 'sunny'/'rain'
'laser_veh_count'	Integer number, the number of vehicles detected by Lidar in current frame
'obj_type'	String, 'vehicle'/'bicycle'/'pedestrian'
'obj_id'	'ego' is AV, other ids are detected objects
'global_time_stamp'	Float, Micro seconds since Unix epoch
'local_time_stamp' (s)	Float, Local time from 0s to around 20s
'local_center_x' (m)	Float, local x coordinate of the object center
'local_center_y' (m)	Float, local y coordinate of the object center
'local_center_z' (m)	Float, local z coordinate of the object center
'global_center_x' (m)	Float, global x coordinate of the object center
'global_center_y' (m)	Float, global y coordinate of the object center
'global_center_z' (m)	Float, global z coordinate of the object center
'length' (m)	Float, length of the object
'width' (m)	Float, width of the object
'height' (m)	Float, height of the object
'heading'	Float, global heading for AV, local heading for other objects
'speed_x' (m/s)	Float, speed x of the object
'speed_y' (m/s)	Float, speed y of the object
'accel_x' (m/s2)	Float, acceleration x of the object
'accel_y' (m/s2)	Float, acceleration y of the object
'angular_speed' (rad/s)	Float, angular speed x of the object, only available for AV

References

AHN, S., CASSIDY, M. J. & LAVAL, J. 2004. Verification of a simplified car-following theory. *Transportation Research Part B: Methodological,* 38**,** 431-440.

ALI, Y., HAQUE, M. M., ZHENG, Z., WASHINGTON, S. & YILDIRIMOGLU, M. 2019. A hazard-based duration model to quantify the impact of connected driving environment on safety during mandatory lane-changing. *Transportation research part C: emerging technologies,* 106**,** 113-131.

ALI, Y., ZHENG, Z., HAQUE, M. M., YILDIRIMOGLU, M. & WASHINGTON, S. 2020a. Understanding the discretionary lane-changing behaviour in the connected environment. *Accident Analysis & Prevention,* 137**,** 105463.

ALI, Y., ZHENG, Z., MAZHARUL HAQUE, M., YILDIRIMOGLU, M. & WASHINGTON, S. 2020b. Detecting, analysing, and modelling failed lane-changing attempts in traditional and connected environments. *Analytic Methods in Accident Research,* 28**,** 100138.

BOCK, J., KRAJEWSKI, R., MOERS, T., RUNDE, S., VATER, L. & ECKSTEIN, L. 2019. The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections. *arXiv preprint arXiv:1911.07602*.

CAESAR, H., BANKITI, V., LANG, A. H., VORA, S., LIONG, V. E., XU, Q., KRISHNAN, A., PAN, Y., BALDAN, G. & BEIJBOM, O. nuscenes: A multimodal dataset for autonomous driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11621-11631.

CHANG, M.-F., LAMBERT, J., SANGKLOY, P., SINGH, J., BAK, S., HARTNETT, A., WANG, D., CARR, P., LUCEY, S. & RAMANAN, D. Argoverse: 3d tracking and forecasting with rich maps. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019. 8748-8757.

CHEN, D., AHN, S., LAVAL, J. & ZHENG, Z. 2014. On the periodicity of traffic oscillations and capacity drop: The role of driver characteristics. *Transportation Research Part B: Methodological,* 59**,** 117-136.

CHEN, D., LAVAL, J., ZHENG, Z. & AHN, S. 2012a. A behavioral car-following model that captures traffic oscillations. *Transportation Research Part B: Methodological,* 46**,** 744-761.

CHEN, D., LAVAL, J. A., AHN, S. & ZHENG, Z. 2012b. Microscopic traffic hysteresis in traffic oscillations: A behavioral perspective. *Transportation Research Part B: Methodological,* 46**,** 1440-1453.

CIUFFO, B., PUNZO, V. & TORRIERI, V. 2008. Comparison of Simulation-Based and Model-Based Calibrations of Traffic-Flow Microsimulation Models. *Transportation Research Record: Journal of the Transportation Research Board,* 2088**,** 36-44.

COIFMAN, B. & LI, L. 2017. A critical evaluation of the Next Generation Simulation (NGSIM) vehicle trajectory dataset. *Transportation Research Part B: Methodological,* 105**,** 362-377.

COIFMAN, R. R. & DONOHO, D. L. 1995. Translation-invariant de-noising. *Wavelets and statistics.* Springer.

COLOMBARONI, C. & FUSCO, G. 2013. Artificial Neural Network Models for Car Following: Experimental Analysis and Calibration Issues. *Journal of Intelligent Transportation Systems,* 18**,** 5-16.

DONOHO, D. L. 1995. De-noising by soft-thresholding. *IEEE transactions on information theory,* 41**,** 613-627.

DONOHO, D. L. & JOHNSTONE, I. M. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association,* 90**,** 1200-1224.

DONOHO, D. L., JOHNSTONE, I. M., KERKYACHARIAN, G. & PICARD, D. 1995. Wavelet shrinkage: asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological),* 57**,** 301-337.

DURET, A., BUISSON, C. & CHIABAUT, N. 2008. Estimating Individual Speed-Spacing Relationship and Assessing Ability of Newell's Car-Following Model to Reproduce Trajectories. *Transportation Research Record: Journal of the Transportation Research Board,* 2088**,** 188-197.

GEIGER, A., LENZ, P., STILLER, C. & URTASUN, R. 2013. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research,* 32**,** 1231-1237.

HAMDAR, S. H. & MAHMASSANI, H. S. 2008. Driver car-following behavior: From discrete event process to continuous set of episodes.

HE, Z., ZHENG, L. & GUAN, W. 2015. A simple nonparametric car-following model driven by field data. *Transportation Research Part B: Methodological,* 80**,** 185-201.

HUANG, X., SUN, J. & SUN, J. 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transportation Research Part C: Emerging Technologies,* 95**,** 346-362.

KESTEN, R., USMAN, M., HOUSTON, J., PANDYA, T., NADHAMUNI, K., FERREIRA, A., YUAN, M., LOW, B., JAIN, A. & ONDRUSKA, P. 2019. Lyft level 5 av dataset 2019. *urlhttps://level5. lyft. com/dataset*.

KRAJEWSKI, R., BOCK, J., KLOEKER, L. & ECKSTEIN, L. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018. IEEE, 2118-2125.

LAVAL, J. A. 2011. Hysteresis in traffic flow revisited: An improved measurement method. *Transportation Research Part B: Methodological,* 45**,** 385-391.

LAVAL, J. A., TOTH, C. S. & ZHOU, Y. 2014. A parsimonious model for the formation of oscillations in car-following models. *Transportation Research Part B: Methodological,* 70**,** 228-238.

LECLERCQ, L., CHIABAUT, N., LAVAL, J. & BUISSON, C. 2007. Relaxation Phenomenon after Lane Changing. *Transportation Research Record: Journal of the Transportation Research Board,* 1999**,** 79-85.

LEE, G. R., GOMMERS, R., WASELEWSKI, F., WOHLFAHRT, K. & O'LEARY, A. 2019. PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software,* 4**,** 1237.

LI, L., JIANG, R., HE, Z., CHEN, X. & ZHOU, X. 2020. Trajectory data-based traffic flow studies: A revisit. *Transportation Research Part C: Emerging Technologies,* 114**,** 225-240.

MONTANINO, M. & PUNZO, V. 2013. Making NGSIM Data Usable for Studies on Traffic Flow Theory. *Transportation Research Record: Journal of the Transportation Research Board,* 2390**,** 99-111.

MONTANINO, M. & PUNZO, V. 2015. Trajectory data reconstruction and simulation-based validation against macroscopic traffic patterns. *Transportation Research Part B: Methodological,* 80**,** 82-106.

MORIDPOUR, S., MAZLOUMI, E. & MESBAH, M. 2015. Impact of heavy vehicles on surrounding traffic characteristics. *Journal of Advanced Transportation,* 49**,** 535-552.
NASON, G. 2008. *Wavelet methods in statistics with R*, Springer Science & Business Media.
NGSIM. 2016. *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data* [Online]. Available: http://doi.org/10.21949/1504477 [Accessed 2020.10.19].
NHTSA. 2021. *Automated Vehicles for Safety* [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety [Accessed 2021.04.08].
OSORIO, C. & PUNZO, V. 2019. Efficient calibration of microscopic car-following models for large-scale stochastic network simulators. *Transportation Research Part B: Methodological,* 119**,** 156-173.
OSSEN, S. & HOOGENDOORN, S. P. 2008. Validity of Trajectory-Based Calibration Approach of Car-Following Models in Presence of Measurement Errors. *Transportation Research Record: Journal of the Transportation Research Board,* 2088**,** 117-125.
OSSEN, S. & HOOGENDOORN, S. P. 2011. Heterogeneity in car-following behavior: Theory and empirics. *Transportation Research Part C: Emerging Technologies,* 19**,** 182-195.
PUNZO, V., BORZACCHIELLO, M. T. & CIUFFO, B. 2011. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation (NGSIM) program data. *Transportation Research Part C: Emerging Technologies,* 19**,** 1243-1262.
PUNZO, V., FORMISANO, D. J. & TORRIERI, V. 2005. Nonstationary Kalman Filter for Estimation of Accurate and Consistent Car-Following Data. *Transportation Research Record: Journal of the Transportation Research Board,* 1934**,** 2-12.
PUNZO, V., MONTANINO, M. & CIUFFO, B. 2015. Do We Really Need to Calibrate All the Parameters? Variance-Based Sensitivity Analysis to Simplify Microscopic Traffic Flow Models. *IEEE Transactions on Intelligent Transportation Systems,* 16**,** 184-193.
PUNZO, V., ZHENG, Z. & MONTANINO, M. 2021. About calibration of car-following dynamics of automated and human-driven vehicles: Methodology, guidelines and codes. *Transportation Research Part C: Emerging Technologies,* 128**,** 103165.
RAFATI FARD, M., SHARIAT MOHAYMANY, A. & SHAHRI, M. 2017. A new methodology for vehicle trajectory reconstruction based on wavelet analysis. *Transportation Research Part C: Emerging Technologies,* 74**,** 150-167.
SAIFUZZAMAN, M. & ZHENG, Z. 2014. Incorporating human-factors in car-following models: A review of recent developments and research needs. *Transportation Research Part C: Emerging Technologies,* 48**,** 379-403.
SAIFUZZAMAN, M., ZHENG, Z., HAQUE, M. M. & WASHINGTON, S. 2017. Understanding the mechanism of traffic hysteresis and traffic oscillations through the change in task difficulty level. *Transportation Research Part B: Methodological,* 105**,** 523-538.
SHARMA, A. & ZHENG, Z. 2021. Connected and Automated Vehicles: Opportunities and Challenges for Transportation Systems, Smart Cities, and Societies. *Automating Cities***,** 273-296.

SHARMA, A., ZHENG, Z. & BHASKAR, A. 2018. A pattern recognition algorithm for assessing trajectory completeness. *Transportation Research Part C: Emerging Technologies,* 96**,** 432-457.

SHARMA, A., ZHENG, Z. & BHASKAR, A. 2019a. Is more always better? The impact of vehicular trajectory completeness on car-following model calibration and validation. *Transportation Research Part B: Methodological,* 120**,** 49-75.

SHARMA, A., ZHENG, Z., BHASKAR, A. & HAQUE, M. M. 2019b. Modelling car-following behaviour of connected vehicles with a focus on driver compliance. *Transportation research part B: methodological,* 126**,** 256-279.

SHARMA, A., ZHENG, Z., KIM, J., BHASKAR, A. & HAQUE, M. M. 2019c. Estimating and Comparing Response Times in Traditional and Connected Environments. *Transportation Research Record: Journal of the Transportation Research Board,* 2673**,** 674-684.

STEPHANE, M. 1999. A wavelet tour of signal processing. *The Sparse Way*.

SUN, P., KRETZSCHMAR, H., DOTIWALLA, X., CHOUARD, A., PATNAIK, V., TSUI, P., GUO, J., ZHOU, Y., CHAI, Y. & CAINE, B. Scalability in perception for autonomous driving: Waymo open dataset. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 2446-2454.

TASWELL, C. 2000. The what, how, and why of wavelet shrinkage denoising. *Computing in science & engineering,* 2**,** 12-19.

THIEMANN, C., TREIBER, M. & KESTING, A. 2008. Estimating Acceleration and Lane-Changing Dynamics from Next Generation Simulation Trajectory Data. *Transportation Research Record: Journal of the Transportation Research Board,* 2088**,** 90-101.

TIAN, J., JIANG, R., JIA, B., GAO, Z. & MA, S. 2016. Empirical analysis and simulation of the concave growth pattern of traffic oscillations. *Transportation Research Part B: Methodological,* 93**,** 338-354.

TOLEDO, T., KOUTSOPOULOS, H. N. & AHMED, K. I. 2007. Estimation of Vehicle Trajectories with Locally Weighted Regression. *Transportation Research Record: Journal of the Transportation Research Board,* 1999**,** 161-169.

TORDEUX, A., LASSARRE, S. & ROUSSIGNOL, M. 2010. An adaptive time gap car-following model. *Transportation Research Part B: Methodological,* 44**,** 1115-1131.

TREIBER, M. & KANAGARAJ, V. 2015. Comparing numerical integration schemes for time-continuous car-following models. *Physica A: Statistical Mechanics and its Applications,* 419**,** 183-195.

TREIBER, M. & KESTING, A. 2013a. Microscopic Calibration and Validation of Car-Following Models – A Systematic Approach. *Procedia - Social and Behavioral Sciences,* 80**,** 922-939.

TREIBER, M. & KESTING, A. 2013b. Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*.

VIEIRA DA ROCHA, T., LECLERCQ, L., MONTANINO, M., PARZANI, C., PUNZO, V., CIUFFO, B. & VILLEGAS, D. 2015. Does traffic-related calibration of car-following models provide accurate estimations of vehicle emissions? *Transportation Research Part D: Transport and Environment,* 34**,** 267-280.

YEO, H. & SKABARDONIS, A. 2009. Understanding Stop-and-go Traffic in View of Asymmetric Traffic Theory. 99-115.

YI, H., EDARA, P. & SUN, C. 2014. Modeling Mandatory Lane Changing Using Bayes Classifier and Decision Trees. *IEEE Transactions on Intelligent Transportation Systems,* 15**,** 647-655.

YU, F., CHEN, H., WANG, X., XIAN, W., CHEN, Y., LIU, F., MADHAVAN, V. & DARRELL, T. BDD100K: A diverse driving dataset for heterogeneous multitask learning. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 2636-2645.

ZHENG, Z. 2014. Recent developments and research needs in modeling lane changing. *Transportation research part B: methodological,* 60**,** 16-32.

ZHENG, Z., AHN, S., CHEN, D. & LAVAL, J. 2011a. Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations. *Transportation Research Part B: Methodological,* 45**,** 372-384.

ZHENG, Z., AHN, S., CHEN, D. & LAVAL, J. 2011b. Freeway traffic oscillations: Microscopic analysis of formations and propagations using Wavelet Transform. *Transportation Research Part B: Methodological,* 45**,** 1378-1388.

ZHENG, Z., AHN, S., CHEN, D. & LAVAL, J. 2013. The effects of lane-changing on the immediate follower: Anticipation, relaxation, and change in driver characteristics. *Transportation Research Part C: Emerging Technologies,* 26**,** 367-379.

ZHENG, Z. & WASHINGTON, S. 2012. On selecting an optimal wavelet for detecting singularities in traffic and vehicular data. *Transportation Research Part C: Emerging Technologies,* 25**,** 18-33.