Overbounding Multipath Error in Urban Canyon With LSTM Using Multi-Sensor Features

Ruirui Liu and Yiping Jiang

***Abstract*— Multipath error is a major challenge for positioning integrity monitoring in autonomous driving, which requires conservative and effective overbounding methods. Traditional methods based on Gaussian overbound need the error distribution information, which is hard to obtain when using multiple features related to multipath as prior knowledge. A recent study used quantile overbound based on multi-layer perceptron (MLP) network with promising results but ignored the temporal correlation of multipath and the dynamic surrounding information. This paper proposes a multipath overbounding method based on a designed long short-term memory (LSTM) network using LiDAR, cameras, and global navigation satellite systems data. The method aims to improve the model's effectiveness while ensuring the model's conservatism. The effectiveness of the model is assessed using a generalized coefficient of determination, which shows how close the predicted quantile is to the actual value. Results show that the LSTM model outperforms the previous MLP-based study in predicting the quantile of multipath error in Hong Kong urban data. By using multi-sensor data as input, the effectiveness of LSTM improves by over 10% when using time windows of 15 seconds for different urban scenarios. The longer the time window, the better the performance of effectiveness. The predicted quantile is then used to compute an overbounded standard deviation based on a zero-mean Gaussian distribution, whose conservatism is verified by the cumulative distribution function. Overall, this study indicates that the use of multi-sensor data and a longer time window can better facilitate the effective bounding of the multipath error while ensuring conservatism.**

***Index Terms*— GNSS, multipath, integrity, deep learning, LSTM.**

I. Introduction

AUTONOMOUS driving is a promising technology that relies on high-integrity localization solutions to ensure user safety. In order to achieve this goal, it is necessary to monitor the integrity of the localization component, which may contain various types of errors resulting in hazardous position errors. protection level (PL) is a statistical upper bound on the position error, while alert limit (AL) is maximum allowable of the position error [1], [2]. An alarm will be triggered when the PL is greater than AL. The integrity risk (IR) is defined as the probability when the position error surpasses the given AL without users' awareness [3], [4]. For autonomous driving in urban environments, a preliminary requirement for integrity risk has been established as 10^{-8}/hour [5]. In comparison, for en-route and precision approach users in civil aviation, the integrity risk requirements are 10^{-7}/hour and 10^{-7}/150sec, respectively. This suggests that the IR for autonomous driving may be even more stringent than that in civil aviation. Additionally, the AL for autonomous driving is typically confined to a specific lane, whereas in civil aviation, the AL has a broader range. Consequently, the PL for autonomous driving needs to be even more rigorous than that for civil aviation [5], [6].

The PL in the position domain is derived from combining the bounding standard deviations of all error sources in the range domain [7], [8]. Typically, measurement errors are characterized by zero-mean Gaussian distributions. However, the fitted standard deviation may not represent the error distributions properly. When error distributions exhibit heavy tails, the fitted Gaussian distributions would assign a lower probability to large errors than the actual error distributions. To ensure conservatism for integrity, an overbounding distribution model is introduced to replace the error distribution. Specifically, the accumulated tail probability of the overbound distribution should exceed that of the measurement error distributions to guarantee conservatism. The conservatism can be transferred by linear operations to ensure that the PL is the upper bound of the position error (PE) [8], [9], [10], [11], [12]. Generally, the overbounding model used the zero-mean Gaussian distribution [8]. However, there are instances where the standard deviation of the overbounding model is significantly larger than that of the actual error distribution. As a result, the PL based on the overbounding standard deviation can be overly conservative, potentially leading to a decline in system availability [8], [13], [14]. Therefore, the process of overbounding requires finding the right balance between the model's conservatism and its effectiveness [14], especially for autonomous driving with strict requirements. The effectiveness of modeling is defined as the model's goodness of fit. In summary, the overbound model should strive to maximize effectiveness while ensuring its conservatism to obtain an appropriate PL.

Overbounding multipath errors, the major source of error for global navigation satellite systems (GNSS) positioning in urban areas [2], [15], is crucial for ensuring the integrity of autonomous driving systems in urban environments. The multipath error occurs when the receiver simultaneously receives

Manuscript received 26 March 2023; revised 20 December 2023; accepted 24 February 2024. Date of publication 27 March 2024; date of current version 29 August 2024. The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. 25202520; 15214523) and the National Natural Science Foundation of China (Project No. 42004029). The Associate Editor for this article was P. Ye. *(Corresponding author: Yiping Jiang.)*

The authors are with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong, China (e-mail: yiping.jiang@polyu.edu.hk).

Digital Object Identifier 10.1109/TITS.2024.3376812

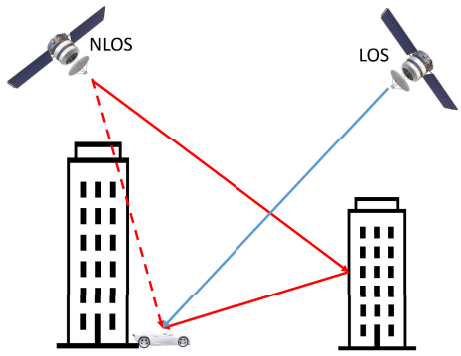

Fig. 1. LOS vs NLOS.

satellite signals from multiple paths. One of these signals comes directly from the satellite without any obstruction and is known as the line-of-sight (LOS) signal. The other signals are reflected, scattered, or diffracted by objects around the receiver, which can distort the amplitude and phase of the direct-path signal, resulting in measurement errors. When the satellite and receiver are completely blocked, and only signals reflected or scattered from surroundings exist, the error produced by these signals is called non-line-of-sight (NLOS) error as depicted in Figure 1. Therefore, multipath errors with LOS signal and NLOS errors exhibit distinct causes and characteristics [16]. To be more specific, the NLOS errors have a larger standard deviation on the order of hundreds of meters than the multipath errors with LOS signal [16], [17]. Some literature distinguishes NLOS errors and the multipath errors with LOS signal, while others collectively refer to them as multipath errors [2], [17]. In the current context, the term "multipath errors" encompasses both multipath errors with LOS signal and NLOS errors.

Traditional methods for overbounding multipath errors have typically employed Gaussian overbound techniques, such as cumulative distribution function (CDF) overbound, paired overbound, and two-step Gaussian overbound [10], [18], [19]. The CDF overbound is applied in [20] to model the code and carrier measurements' multipath errors, and two-step Gaussian overbound is used to model multipath errors [14]. However, the traditional methods model multipath errors on total error distributions without prior knowledge or they rely solely on elevation angle as prior knowledge by dividing data based on elevation angle intervals, which has limited prior knowledge and can lead to an overly conservative model. If more relevant prior knowledge is provided for overbounding multipath errors, the model will be more effective while still ensuring conservatism. However, when introducing more than one feature as prior knowledge for overbounding multipath errors, it becomes challenging to determine the joint probability density function (PDF) and conditional distribution, especially for continuous variable features [17], [21]. This limitation renders traditional Gaussian overbound methods inapplicable. To address this issue, the quantile overbound method has been proposed. This method leverages the multilayer perceptron (MLP), a type of neural network in machine learning, to model multipath errors and predict the quantile of the conditional distribution of multipath errors. This prediction is then converted into an overbounding standard deviation [22]. Neural networks are a viable method for this purpose, based on the universal approximation theorem. This theorem states that neural networks of arbitrary width or depth have the capability to represent all continuous function [23]. The MLP model is trained using features obtained from a single epoch of GNSS measurements, where an "epoch" refers to the duration of the GNSS measurement update interval [22]. However, this approach does not account for the temporal correlation of multipath errors over time. Previous studies have conducted sufficient research on the temporal correlation of multipath errors. It has been found that for a static receiver in an open-sky environment, the multipath errors exhibit a significant cycle with a time constant greater than 600 seconds [20], [24], [25], [26]. On the other hand, for a dynamic receiver in a complex environment, the multipath errors have a smaller time constant [20]. Additionally, the MLP model in [22] only selects GNSS measurement features (carrier-to-noise-density ratio (CN0) and elevation) and a topographic database. However, the topographic database only considers the static surrounding information, failing to incorporate the dynamic surrounding details, such as surrounding vehicles that can also contribute to the multipath errors in the urban environment [20]. Furthermore, there is no method provided in [22] to quantify the effectiveness of the overbounding model. To address the limitations mentioned, this study proposes an overbounding method for multipath errors using a designed long short-term memory (LSTM) network with multi-sensor input. LSTM is a particular form of recurrent neural network (RNN) that performs well in capturing temporal correlations, making it a great success in sequence tasks, such as language translation and signal processing. To capture both static and dynamic information, this study employs features from multiple sensors including camera, LiDAR, and GNSS as input for the LSTM. The LSTM network generates the quantile of the conditional distribution based on the temporal correlated prior knowledge provided by the various sensor features. The network is trained using designed Huber quantile loss function, an extension of the classical Huber loss function [27]. Subsequently, an overbounding standard deviation is calculated using the quantile overbound method. The effectiveness of the overbounding model is measured using a modified coefficient of determination $\mathscr{R}$, while conservatism is verified by plotting the CDF of normalized multipath errors and standard Gaussian distribution.

This paper is organized as follows. Section II outlines the details of the proposed overbounding method for multipath errors, including the overbounding framework architecture, multipath estimation, feature extraction, LSTM network modeling, quantile overbound, and evaluation. Subsequently, experiments are conducted using different realistic urban scenarios in Hong Kong to validate the proposed method and analyze the effectiveness of the method, also in comparison with the benchmark in Section III. Finally, based on the outcomes analyzed in III, the proposed method is discussed, highlighting its advantages and future work in Section IV.

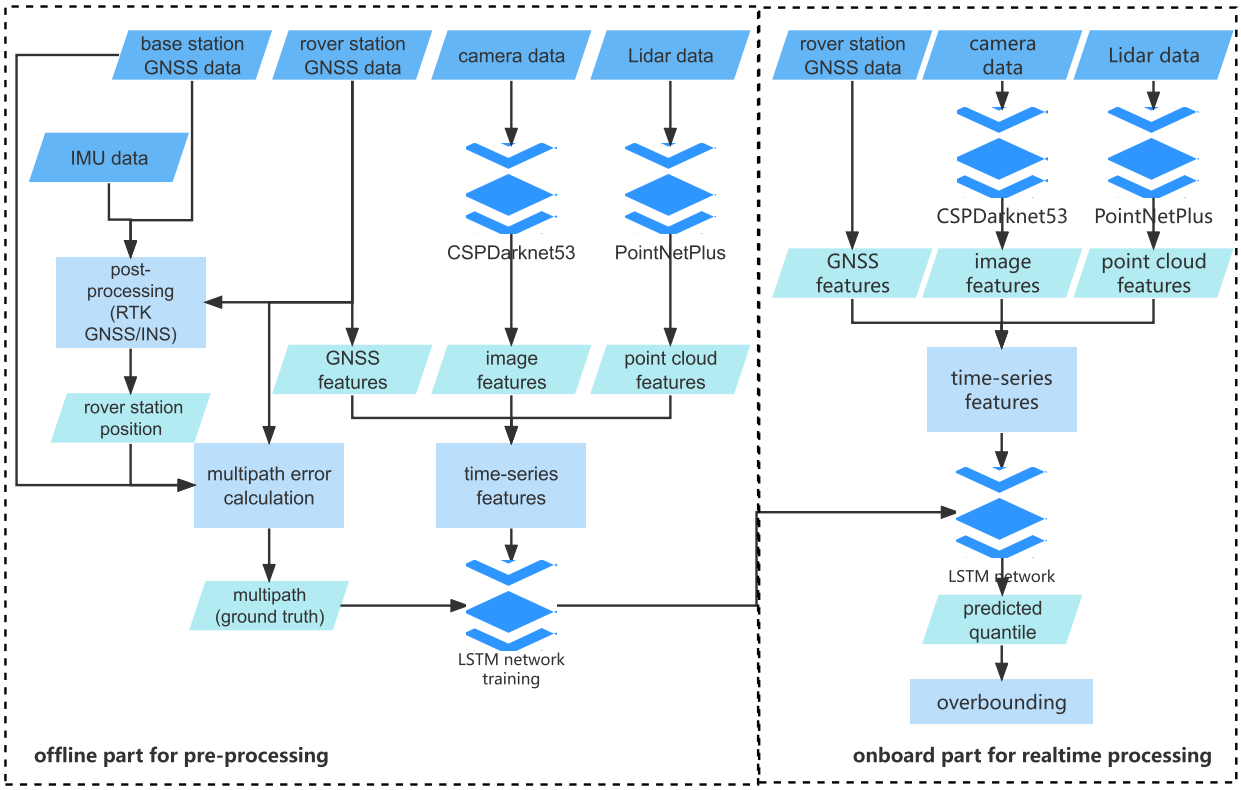

Fig. 2. The architecture of data processing.

II. Modelling Methodology

This section presents a detailed description of the overbounding framework architecture, followed by discussions on multipath estimation, feature extraction, LSTM network modeling, quantile overbound, and evaluation. The framework is divided into two parts: the offline component for pre-processing and the onboard component for real-time processing. The offline segment estimates the multipath errors and models the multipath errors via sensor measurements using the LSTM network. On the other hand, the onboard segment generates the quantile of multipath error distribution and the bounded standard deviation of multipath errors for protection level calculation. The framework architecture is depicted in figure 2.

The offline part is performed with the following steps:

1) Obtaining the precise position of the vehicle (rover station) from Novatel product, SPAN-CPT receiver using real-time kinematic positioning (RTK) integrated with an inertial measurement unit (IMU);
2) Estimating the multipath errors as ground truth for network training, as described in detail in the section II-A;
3) Extracting temporal correlated image and point cloud features from camera and LiDAR data using pretrained deep neural networks and fusing them with GNSS features, as explained in the section II-B3;
4) Training the designed LSTM network using the time-series features and the multipath errors. The network details are provided in the section II-B.

For onboard real-time processing, the following steps are performed:

1) Extracting time-series features from multi-sensor data, similar to Step 3 in the offline part;
2) Predicting the quantile of the conditional distribution by inputting the time-series features into the trained LSTM network from Step 4 in the offline part;
3) Obtaining the bounding standard deviation using quantile overbound, as expressed in the section II-B

A. Multipath Estimation

This study focuses on multipath errors in pseudorange measurement. To estimate the multipath errors, one common method uses the ionosphere-free code-minus-carrier (CMC) [8], [14], [20], but it does not consider the NLOS error. Another method utilizes the single differenced (SD) measurement between two satellites to estimate the multipath errors [28]. In addition, assuming that the multipath errors at the base station are negligible compared with the user receiver, the double differenced (DD) method can be used for estimating the multipath errors [29]. Both DD and SD methods assume the multipath errors from the satellite with the highest elevation angle are negligible, which may be inaccurate in urban environments. As an alternative method, the SD pseudorange measurement between the base station and rover station and its rate of change are used to estimate the difference receiver clock and clock rate with Kalman filter to obtain the multipath errors [22], which is adopted for this study. Specifically, the state update equation and the dynamic model equation in the Kalman filter are expressed as [22]:

$$x_k = \begin{bmatrix} c\Delta t_k \\ c\Delta \dot{t}_k \end{bmatrix} = \begin{bmatrix} c\Delta t_{k-1} + \delta t c\Delta \dot{t}_{k-1} \\ c\Delta \dot{t}_k \end{bmatrix} + w_k$$

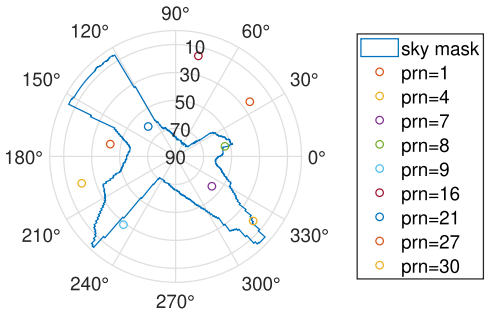

Fig. 3. Sky mask example.

$$= \begin{bmatrix} 1 & \delta t \\ 0 & 1 \end{bmatrix} x_{k-1} + w_k \tag{1}$$

$$z_k = \begin{bmatrix} P_r^s - P_b^s - (\hat{\rho}_r^s - \hat{\rho}_b^s) \\ \dot{P}_r^s - \dot{P}_b^s - (\hat{\dot{\rho}}_r^s - \hat{\dot{\rho}}_b^s) \end{bmatrix} = \begin{bmatrix} c\Delta t_k + \Delta\epsilon_k \\ c\Delta \dot{t}_k + \Delta\dot{\epsilon}_k \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_k + v_k \tag{2}$$

where z_k denotes the measurement at epoch k; x_k is the state at epoch k; v_k and w_k are the measurement noise and process noise; c is the speed of light; Δt represents the slowly varying bias error, including the residual hardware bias of receiver stations and the residual receiver clock error; δt is time interval between two adjacent epochs; $c\Delta\dot{t}$ contains the clock drift; P is pseudorange measurement; superscript s represents the satellite, subscript r denotes the rover station, b represents the base station; ρ is the satellite-to-receiver range and is calculated using the satellite precise orbit from the precise orbit and clock product, and the receiver precise position obtained from the Novatel SPAN-CPT receiver's output; ϵ is multipath errors and noise (MPN); $\Delta\epsilon$ represents the rover station's MPN, neglecting the residual of ionosphere delay, troposphere delay, and base station's MPN; $\Delta\dot{\epsilon}$ represents the change rate of $\Delta\epsilon$.

Since the multipath errors are elevation angle dependent and NLOS error has a larger variance than the LOS multipath error [16], [17], [30]. In contrast to the approach that utilizes a unified standard deviation for all satellites in [22], this article distinguishes between NLOS and LOS signals for each satellite by employing sky masks and assigns different standard deviation to each satellite based on its elevation angle. The sky masks serve to define the boundaries of the surrounding buildings at a specific location, which is derived from a 3D model or sky-view images [31]. Figure 3 provides a visual representation of a sky mask, where the blue boundary represents its extent. The sky-view images are obtained through the simulation of a virtual fisheye camera using Google Earth Studio [32].

The elevation-angle-dependent weight matrix is designed for the covariance matrix of the measurement noise, given by:

$$\Sigma(v) = diag((\frac{E(sin(elv))}{sin(elv)}\sigma_{MPN})^2) \tag{3}$$

where

$$\sigma_{MPN} = \begin{cases} \sigma_{LOS}, & \text{satellite is in the sky mask} \\ \sigma_{NLOS}, & \text{satellite is out of the sky mask} \end{cases}$$

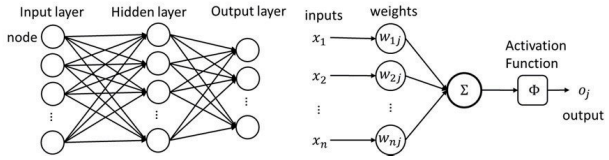

Fig. 4. Neural network structure (left) and neural network node structure (right).

where σ_{LOS} and σ_{NLOS} are two constants that can be set based on the field environment. As depicted in Figure 3, the blue boundary represents the sky mask. When a satellite falls within the sky mask, it predominantly exhibits multipath errors with LOS signal, and the corresponding σ_{LOS} is employed. Conversely, if the satellite is located outside the sky mask, NLOS error becomes the primary error, and σ_{NLOS} is utilized. The expectation of elevation angles at all epochs and satellites in a scenario is also a constant denoted as $E(sin(elv))$. Section III-B demonstrates the rationality for the varying covariance matrix method using real-world data.

B. Multipath Modelling With Designed LSTM Network

1) Deep Learning Background: Neural networks comprise an input layer, one or more hidden layers, and an output layer, shown in the left part of figure 4. Each Layer is constructed with nodes, and each node connects to others with an associated weight and bias. The right part of figure 4 further illustrates the detail of each node. The node's input passes through the non-linear activation function to determine if the node is activated, like a neuron.

The hidden layer has various types, such as convolutional, dense, and recurrent layers. In this paper, the convolutional layer designed for two-dimensional or three-dimensional data is used for image feature extraction. The dense layer refers to each node in one layer being connected to all nodes in the next layer. The MLP used in [22] and article consist of several dense layers. RNN is a widely used neural network containing recurrent layers, which can keep and pass information through the same layer. Specifically, RNN contains cyclical connections that allow the network to incorporate information from previous steps into the prediction made at the current step. The activations from a previous step are fed as inputs into the network for prediction at the current step, making it naturally suitable for time-series tasks [33], as shown in the left part of figure 5. However, if the predictions are highly related to a relatively far time step, the performance of RNN is degraded due to limited information, which is the so-called long-term dependency issue. The LSTM network is a particular form of RNN capable of dealing with the long-term dependency issue [34], [35], [36]. The right part of figure 5 shows the structure of a single LSTM cell (node) with three gates, including the input gate i_t, the forget gate f_t, and the output gate o_t [34].

The forget gate determines the information that should be neglected from the last step, and the input gate determines which part of the information from input data should remain. The output gate determines the output information and the

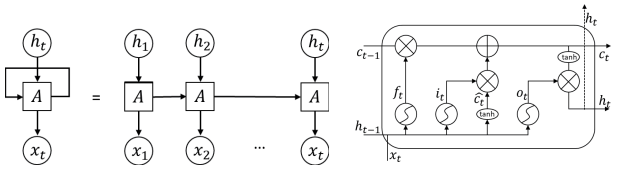

Fig. 5. LSTM model Structure (left) and LSTM cell (node) structure (right).

remaining information for the next time step. The mathematical model of the LSTM cell can be analytically expressed by following equations [37],

$$f_t = \sigma(w_f(h_{t-1}, x_t) + b_f) \tag{4}$$
$$i_t = \sigma(w_i(h_{t-1}, x_t) + b_i) \tag{5}$$
$$\hat{c_t} = tanh(w_c(h_{t-1}, x_t) + b_C) \tag{6}$$
$$c_t = f_t c_{t-1} + i_t \hat{c_t} \tag{7}$$
$$o_t = \sigma(w_o(h_{t-1}, x_t) + b_o) \tag{8}$$
$$h_t = o_t * tanh(c_t) \tag{9}$$

where x_t is the cell input at epoch t, combined with the output of a cell h_{t-1} at the previous epoch $t-1$, the combined vector passes through forget, input, and output gates. Then, the forget gate determines whether to keep cell data c_{t-1} at the previous epoch $t-1$, c_t is the current cell memory state, and h_t is the output of the cell.

2) Designed Neural Network Architecture: Figure 6 presents the neural network architecture developed in this study. The architecture can be divided into two parts, the left part of the figure 6 represents the feature extractor component, which processes the sensor data from the vehicle, including GNSS, camera, and LiDAR data. Each type of data has a different feature extractor. The feature extracted from GNSS data contains the GNSS signal characteristics. The features extracted from camera and LiDAR data encompass the surrounding features of the vehicle, such as surrounding objects, surrounding surface, and surrounding geometric relationship. The feature extractors for camera and LiDAR data consist of two parts: state-of-the-art feature extractors and extra feature extractors. The state-of-the-art feature extractor is designed for extracting general surrounding features, while extra feature extractor is designed to extract specific features for our specific task. Details are illustrated in Section II-B3. On the top right part of the figure 6 is a MLP network with four dense layers, used in [22]. The MLP network, named MPNMLP, uses concatenated features from multi-sensor features at one epoch as input to predict the quantile as the benchmark represented the results of [22]. The bottom right part of the figure 6 is the designed network, named MPNLSTM, including an LSTM layer and three dense layers. The sequential GNSS, image, and point cloud features are concatenated over a time-series length τ and then inputted into the MPNLSTM. The network for processing *tau* length is denoted as MPNLSTM_τ. The value of τ in MPNLSTM is set to 5s, 10s, and 15s, to evaluate the influence of time-series length on modeling. Both models, along with the extra feature extractors, are trained through backpropagation of gradients to optimize the weights and biases, with the number of hidden nodes varying based on the number of features n. It should be noted that the batch normalization layer, dropout layer, and activation function are not shown in figure 6.

3) Feature Extractor Component: As depicted in Figure 6, this study utilizes GNSS, camera, and LiDAR data from the vehicle to extract signal-related features and surrounding features in the feature extractor component. Signal-related features: elevation angle, CN0 and azimuth angle, which can be extracted from GNSS data, are highly related to multipath error [8], [14], [26], [30], [38], [39]. In the feature extractor component, these features, denoted as GNSS features, are normalized and trained in the neural network. The surrounding environment also significantly influences multipath error. Camera and LiDAR data provide physical information about the vehicle's surroundings, including the surfaces of buildings, vehicles, pedestrians, and roads. These data sources are used to extract abstract surrounding features, encompassing both dynamic and static features, through the training of neural networks. Dynamic features include semantic abstract features of objects like pedestrians and vehicles, while static features pertain to buildings and lanes. In this study, the image feature extractor is used to derive effective image features, enabling rapid training and optimal performance. The extractor employs the pretrained CSPDARKnet53 model, which was initially trained on the COCO dataset [40]. This dataset is extensive and designed for image recognition and detection tasks. A pretrained model, in essence, is a deep learning model previously trained on a large dataset for a specific task, and can be used as is or customized for different tasks. In this context, a fixed pretrained model is used to extract high-level features from camera data in the semantic space. To extract specific image features associated with multi-path errors, an extra feature extractor, is incorporated into the training process. The extractor consists of three convolution layers with 3×3 kernels. The dimensionality is reduced to a $19 \times 19 \times 1$ feature map, which is subsequently reconstructed into a 361-feature array denoted as $f_{img} = \{f_{img_1}, \cdots, f_{img_{361}}\}$. Moreover, this study uses the pretrained PointNet++ network on the ScanNet dataset for point cloud feature extraction to obtain surrounding semantic features with LiDAR data input. The PointNet++ network is trained to predict point-wise labels for semantic segmentation tasks [41]. The input is 8192 down-sampled points, and the output is a $16 \times 512 \times 1$ feature map in the pretrained backbone. The 10Hz LiDAR data is aligned with the 1Hz GNSS data by averaging the point cloud features across ten epochs. The $16 \times 512 \times 1$ feature map is then fed into an extra feature extractor with several convolution layers and flattened into a 512-feature array, denoted by $f_{pc} = \{f_{pc_1}, \cdots, f_{pc_{512}}\}$.

4) Loss Function: Statistically, minimizing the quantile loss (QL) can obtain the quantile from a distribution [42]. Since the traditional QL function is non-differentiable at zero, and the magnitude of the constantly updated gradient is significant for small errors, it is difficult for the LSTM network to converge to the minimum value by using the traditional QL function. Therefore, a new Huber quantile loss (HQL) function is designed based on the Huber loss function to overcome this problem [27].

The traditional QL function $\mathcal{L}_q^{QL}$ is an asymmetric convex loss function that assigns a weight of q to the overestimation

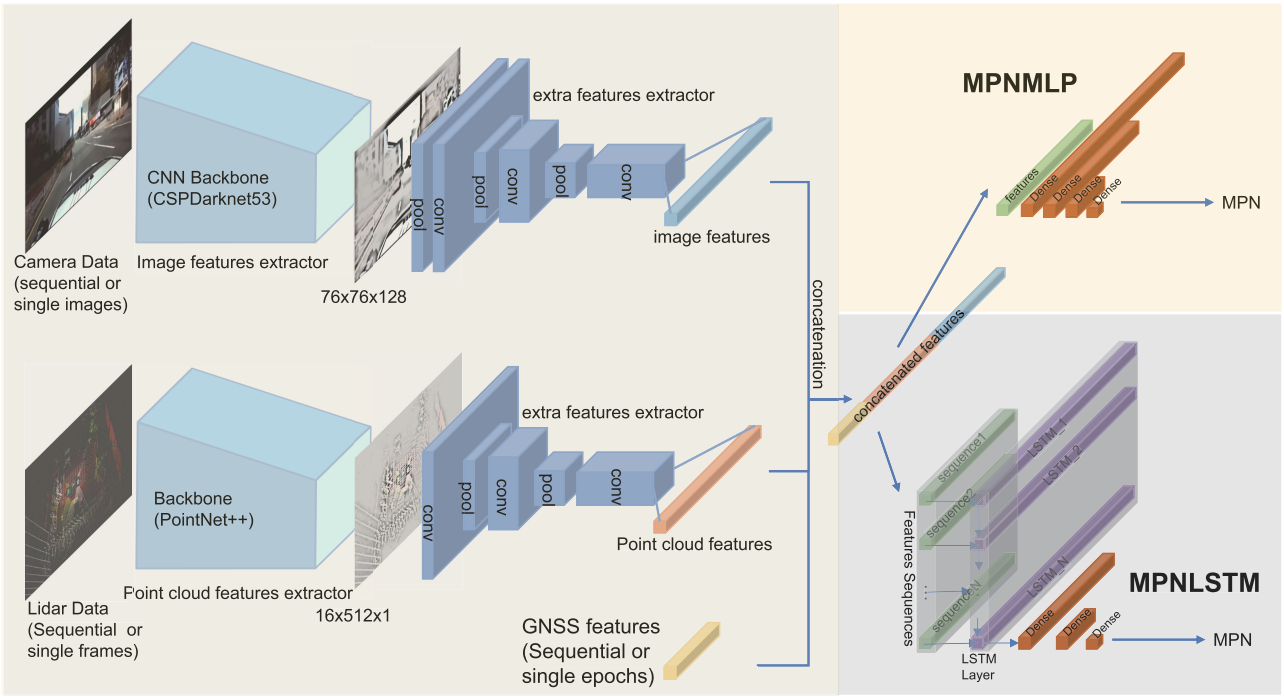

Fig. 6. Neural Network Structure.

of errors and a weight of $1-q$ to the underestimation of errors, where the probability of expected quantile is denoted by $q \in (0,1)$. For a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$, the y_i is from an unknown distribution Z with prior knowledge x_i, denoted as $y_i \sim Z|x_i$ and N is the number of samples, and a given probability q, the quantile of conditional possibility $F^{-1}_{y\sim Z|x}(q)$ can be transformed to find a function $\theta(x)$ to minimize the quantile regression cost function $\mathcal{J}_q^{QL}$, which is the mean of $\mathcal{L}_q^{QL}$.

$$F^{-1}_{y\sim Z|x}(q) = \hat{\theta} = \arg\min_{\theta}(\mathcal{J}_q^{QL}(y, \theta(x))) \tag{10}$$

$$\mathcal{J}_q^{QL}(y, \theta(x)) := E_{y\sim Z|x}[\mathcal{L}_q^{QL}(y-\theta(x))] \tag{11}$$

$$\mathcal{L}_q^{QL}(\epsilon) = \epsilon(q - \delta_\epsilon(\epsilon < 0)), \forall \epsilon \in \mathbb{R} \tag{12}$$

where δ is the Dirac delta function.

And the HQL function is designed as an asymmetric quadratic loss function for $|\epsilon| \leq \sigma$ around zero to make the loss function smooth and linear for $|\epsilon| > \sigma$, where $\sigma \in \mathbb{R}^+$ is a given positive constant. The HQL function is convex and first-order differentiable expressed as

$$\mathcal{L}_q^{HQL}(\epsilon) = \frac{|q_k - \delta_\epsilon(\epsilon < 0)|\mathcal{H}_\sigma(\epsilon)}{\sigma} \tag{13}$$

where $\mathcal{H}_\sigma(\epsilon)$ is the Huber loss function [27], represented as

$$\mathcal{H}_\sigma(\epsilon) = \begin{cases} \frac{1}{2}\epsilon^2, & |\epsilon| \leq \sigma \\ \sigma(|\epsilon| - \frac{1}{2}\sigma), & |\epsilon| > \sigma \end{cases} \tag{14}$$

$q_k = kq$ with k as a zooming factor close to 1. k is used to deal with the quantile zoom issue, where the minimizer of the mean of $\mathcal{L}_q^{HQL}(\epsilon)$ is not precisely equal to the desired quantile of a distribution. This is caused by using the expectation in the interval $[-\sigma, \sigma]$ as the expectation of the quadratic loss function instead of the quantile value. Therefore, k can be derived as:

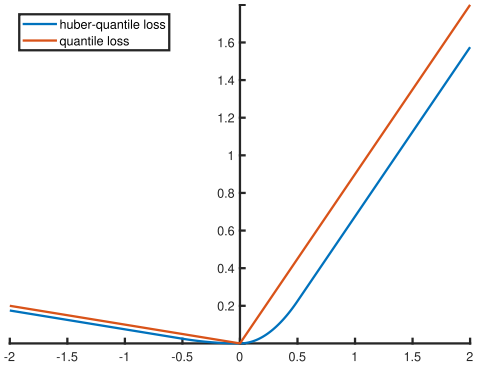

Fig. 7. HQL (blue, $\sigma = 0.5, k = 1$) vs QL (red) for $q = 0.9$.

$$k = \frac{q(1 - e^{-(1-q)\sigma})}{q^2(1 - e^{-(1-q)\sigma}) + (1-q)^2(1 - e^{-q\sigma})} \tag{15}$$

a formal derivation is provided in the Appendix.

The HQL function combines the advantage of the quadratic loss function in computation efficiency and the advantage of the QL function in outlier robustness. In particular, if $\sigma \to 0$, the loss function becomes the QL function. figure 7 compares both loss functions, where the desired probability $q = 0.9, \sigma = 0.5$.

Additionally, an l_2 norm regularization part $\mathscr{L}_{norm} = \frac{\lambda}{2n}\Sigma w_i^2$ is added to the loss to generalize the model to avoid the overfitting issue. Therefore, the objective function is expressed as

$$\mathcal{J}_q^{HQL}(y, \theta(x)) := E_{y\sim Z|x}[\mathcal{L}_q^{HQL}(y-\theta(x))] + \mathscr{L}_{norm} \tag{16}$$

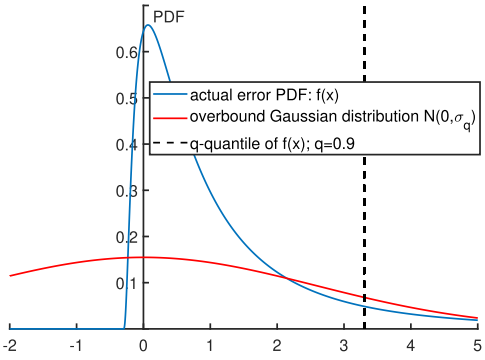

Fig. 8. Quantile overbound description.

Therefore, the θ is obtained by minimizing the objective function. The performance of HQL and QL in a simulation will be presented in Section III-A.

C. Quantile Overbound

This research employs the quantile overbound to estimate a conservative standard deviation by using a zero-mean Gaussian distribution with the same desired quantile as the error distribution. Traditional Gaussian overbound is not suitable for this research, which requires a known error distribution, challenging to obtain from the conditional MPN distribution using LiDAR, camera, and GNSS data as prior knowledge. Figure 8 demonstrates the quantile overbound method.

As shown in Figure 8, the blue line represents the PDF of the actual error distribution, the red line represents the PDF of the overbound model, while the dotted line indicates the quantile value of the actual error at the expected probability of q. Notably, both PDFs share the same quantile value, which enables the derivation of the standard deviation σ_q of the overbound model, given by:

$$\sigma_q = \frac{y}{F_{\mathcal{N}}^{-1}(q)} \tag{17}$$

where y is quantile of the actural error distribution at expected probability q, which can be estimated as $\hat{y} = \theta(x|q)$ in Section II-B; $F_{\mathcal{N}}^{-1}(q)$ is the inverse CDF of Gaussian distribution. Therefore, the overbound model is conservative if the probability that the error exceeds y is not greater than the tail probability of the overbound model.

D. Verification and Evaluation Method

A metric is used for measuring the model's goodness of fit to reflect its effectiveness,

$$\mathscr{R} = 1 - \frac{\mathcal{J}_q^{QL}(y, \theta(x))}{\mathcal{J}_q^{QL}(y, F_{y\sim Z(q)^{-1}})} = 1 - \frac{\mathcal{J}_q^{QL}(y, F_{y\sim Z|x}^{-1}(q))}{\mathcal{J}_q^{QL}(y, F_{y\sim Z}^{-1}(q))} \tag{18}$$

where $F_{y\sim Z}^{-1}(q)$ is the quantile value of total distribution Z, and $\theta(x)$ is determined by model's prediction using input x. From (10) and (11), $\mathcal{J}_q^{QL}(y, \theta(x)) = \mathcal{J}_q^{QL}(y, F_{y\sim Z|x}^{-1}(q))$ is the expectation of quantile loss function using the predicted quantile of conditional possibility. And the $\mathcal{J}_q^{QR}(y, F_{y\sim Z}^{(}-1)(q))$ means the loss without model prediction using the quantile of total distribution Z. If $\mathscr{R} \to 1$, the model has high goodness of fit; and if $\mathscr{R} \leq 0$, the result from the model is worse than the result using the quantile of total distribution Z directly. In this study, x equals the features F corresponding with the multipath errors y.

The overbound models are also checked by plotting CDFs of normalized MPN distribution to evaluate if the model is bounded conservatively. This study uses the quantile overbound method to get the overbound standard deviation in Section II-C. Then the normalized MPN is calculated by $y_{norm} = \frac{y}{\sigma_q}$, and the Gaussian distribution line should overpass the y_{norm} at the probability q when the model is conservative.

III. Simulation and Experimental Results

This section presents a simulation and several experiments. A simulation compares the performance between the HQL and QL functions to illustrate the proposed HQL function suitable for this research. Then, several experiments are conducted on the real-world data in Hong Kong from the UrbanNav dataset to analyze the impact of different sensor features and compare the performance between the proposed LSTM network and the benchmark, MLP in [22]. Furthermore, the models' conservatism is validated, and the models' efficiency is analyzed.

A. Simulation Test for Huber Quantile Loss Vs Quantile Loss

To compare the performance of the modified HQL function and the QL function, a simulation test is conducted. The simulation test is designed using two random variables, $x \sim U(4, 16)$ and $y \sim U(0, 10 * PDF_{\mathcal{N}(x)})$. Here, $U(a, b)$ represents a uniform distribution over the range $[a, b]$, and $PDF_{\mathcal{N}(x)}$ denotes the PDF of a Gaussian distribution. The objective of this simulation is to emulate the modeling of multipath errors in a simplified manner. For each given value of x, the conditional distribution of y follows a Gaussian distribution. The x values are generated using a uniform distribution to ensure an unbiased dataset. A simple network with three dense layers and a designed probability $q = 0.99$ are set up for testing using different loss functions: HQL ($\sigma = 0.5$) and QL. In this context, y represents the ground truth of the model, while x is the input feature. After training the designed network for 3000 epochs, the results are presented in the left part of Figure 9. The simulated data is represented by the blue dot, the red curve represents the predicted value of the quantile using the HQL model, and the orange curve represents the predicted quantile using the QL model.

The dataset $\mathcal{D}$ is randomly separated into a training, validation, and testing dataset with the ratio [0.7,0.15,0.15], respectively. The same training test is repeated 200 times, and the averaged $\mathscr{R}$ results on the testing dataset and all data for the HQL and QL models are shown in Table I and figure 9 right part.

From Table I and figure 9 right part, it can be concluded that the HQL model converges at around 1500 training epochs faster than the QL model at about 2500 training epochs. And

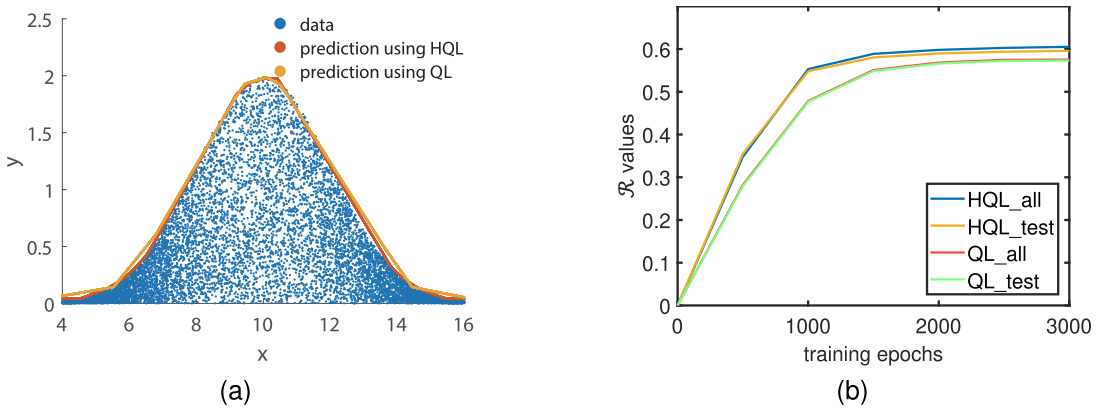

Fig. 9. The scatter plot for data (blue) and predictions using HQL (red) and QL (orange) for q=0.99 (a); Training progress plots of $\mathscr{R}$ metric for HQL and QL model on the testing dataset and all data (b).

TABLE I
$\mathscr{R}$ VALUES USING HQL AND QL MODEL ON THE TESTING DATASET AND ALL DATA FOR DIFFERENT EPOCHS

epochs	HQL_test	HQL_all	QL_test	QL_all
500	0.3552	0.3480	0.2790	0.2820
1000	0.5479	0.5531	0.4765	0.4785
1500	0.5804	0.5887	0.5485	0.5508
2000	0.5897	0.5981	0.5660	0.5684
2500	0.5938	0.6027	0.5723	0.5748
3000	0.5959	0.6052	0.5729	0.5753

before convergence, the HQL performs far better than the QL model with $\mathscr{R}$ values being 0.35 and 0.28 at training epoch 500, and 0.55 and 0.475 at training epoch 1000, respectively. Also, the performance for HQL has a tiny improvement compared to the QL model after convergence. Figure 9 left part illustrates this improvement figuratively, especially at the tail of the distribution. A possible reason is that the data on both sides of the tails are denser than the middle part, resulting in a smaller magnitude of errors between the predicted value and ground truth than the central part. HQL is designed to accommodate such minor errors better than QL. Therefore, the designed HQL converges more efficiently and is more suitable for multipath modeling than the QL model.

B. UrbanNav Dataset and Analysis

This section begins by introducing the dataset used in the experiments, followed by an analysis of MPN distribution in the dataset. Subsequently, a temporal correlation analysis of MPN in the dataset is illustrated. Finally, the modified estimated MPN method demonstrates its rationality. The UrbanNav dataset applied in this study uses a Honda electric car equipped as figure 10 with the following sensors to collect raw GNSS measurements and other sensor data [43]:

- 3D LiDAR sensor: HDL 32E Velodyne
- Slant LiDARs: VLP16 Velodyne, LsLiDAR C16
- IMU: Xsens Mti, AHRS
- GNSS receivers(x3): u-blox ZED-F9P(x2), EVK-M8T(x1) (1 Hz)
- NovAtel Flexpak6 (1 Hz)
- Camera: ZED2 Stereo (15 Hz)
- precise GNSS receiver: SPAN-CPT

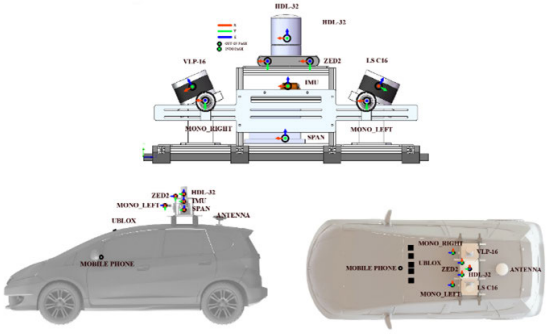

Fig. 10. UrbanNav dataset Hardware Setup [43].

The high-precision estimation of the rover receiver from Novatel SPAN-CPT is regarded as the position ground truth. Three different scenarios are tested including the Hong Kong medium urban, deep urban, and harsh urban trips with detailed descriptions in Table II. In this article, the data from HDL 32E Velodyne Lidar are regarded as Lidar input, and the data from ZED2 stereo camera are image input.

Figure 11 shows the MPN values distribution for the three trips, respectively, expressed the characteristics of multipath errors. All three distributions have heavy tails and positive means, mainly due to NLOS error caused by high buildings in urban areas. Additionally, the error distributions for the deep and harsh urban trips resemble Laplace distributions rather than Gaussian distributions resulting from severe multipath effects causing the error distribution to be more dispersed. The Gaussian distribution is defined in terms of the exponential function of squared difference from the mean, while the Laplace density is expressed in terms of the exponential function of absolute difference from the mean. Consequently, the Laplace distribution has fatter tails than the Gaussian distribution. The error distribution for the harsh urban trip has a longer tail than the deep urban trip, likely due to the more complex urban environment with heavier multipath effects and more NLOS errors. It is interesting to note that the medium urban trip is bimodal with a large bias. One possible reason is that there are fewer GPS satellites with high elevation angles in the medium urban trip compared with the deep and harsh urban trips. Another possible reason is that the duration of the

TABLE II
DATASET DESCRIPTION

	Total Size	Path Length	Sensors	Urban Canyon
UrbanNav-HK-Medium-Urban-1	33.7 GB	3.64 Km	LiDARs/Stereo Camera/IMU/GNSS	Medium
UrbanNav-HK-Deep-Urban-1	63.9 GB	4.51 Km	LiDARs/Stereo Camera/IMU/GNSS	Deep
UrbanNav-HK-Harsh-Urban-1	147 GB	4.86 Km	LiDARs/Stereo Camera/IMU/GNSS	Harsh

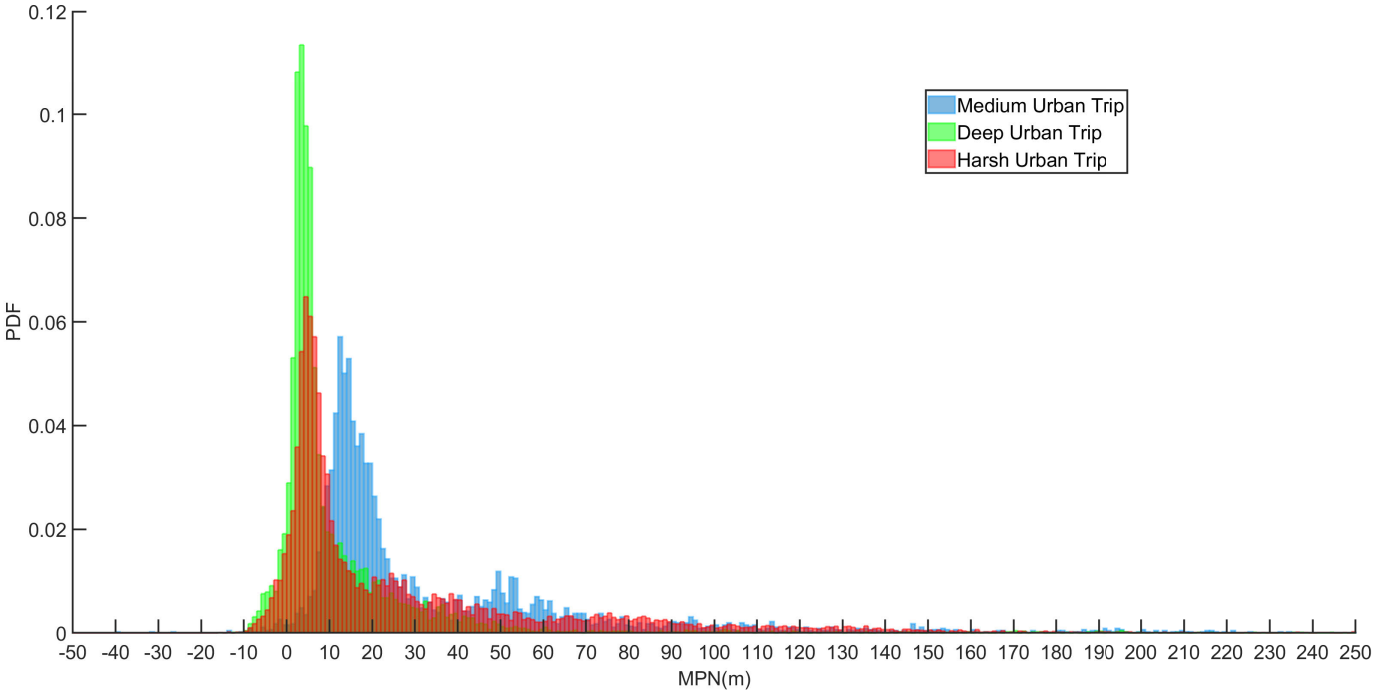

Fig. 11. GPS satellite multipath errors total distribution in HK medium (blue), deep (red), harsh (green) urban trips.

medium urban trip is shorter than the deep and harsh urban trips with fewer data.

Figure 12 provides examples of the autocorrelation function for several satellites in the HK medium urban trip, illustrating the temporal correlation on MPN. The left part of the figure displays the changes in MPN over time, while the right part showcases the autocorrelation function. Additionally, a black horizontal line representing the exp(−1) value is visible. The intersection of this black line with the autocorrelation function indicates the first-order Gauss Markov time constant [20]. It can be observed that the MPN time constant is over 7 seconds, and when considering all the data the time constant of MPN is calculated to be 13.54 seconds.

To illustrate the reasonableness of the newly estimated method in Section II-A, figure 13 shows two sky plots at epoch 66 in the medium urban trip, with the sky masks represented by the blue line. The estimated MPN using a constant covariance matrix is listed in the legend for each satellite in the left figure, while the right figure displays the estimated MPN using a varying covariance matrix as described in Section II-A in the legend. These two figures aim to indicate the improvement of the modified method in this study.

From the sky plots in figure 13, LOS satellites include pseudorandom noise (PRN) code = 7 (purple), 8 (green), and 21 (navy), while all other satellites are NLOS satellites. The left plot shows that the LOS satellites have similar absolute values of estimated MPN around 60 meters, which cannot be clearly distinguished from those of NLOS satellites. The right plot indicates that LOS satellites have minor multipath effects with much smaller errors. Moreover, the multipath errors with a constant covariance matrix results in a negative MPN for an NLOS satellite (PRN = 30), which is inconsistent with reality. Therefore, the method with a varying covariance matrix is more reasonable.

C. Modeling Experiments on Three Trips

The part is divided into four stages:

1) Hardware and software specification;
2) Evaluation of the effectiveness and processing speed on models with various features input;
3) Comparison of the effectiveness between the proposed LSTM model and the benchmark MLP model used in a previous study [22];
4) Analysis of conservatism in the above models.

Each experiment is executed five times with five fixed random seeds to control the random error in the training process resulting from random initialization of the model's parameters and the data shuffling. Furthermore, each trip dataset is separated into a training set and a testing set in the ratio of [0.8, 0.2].

1) Hardware and Software Specification: With the hardware and software specifications listed in Table III, the hardware and software performance is not the best. In practice,

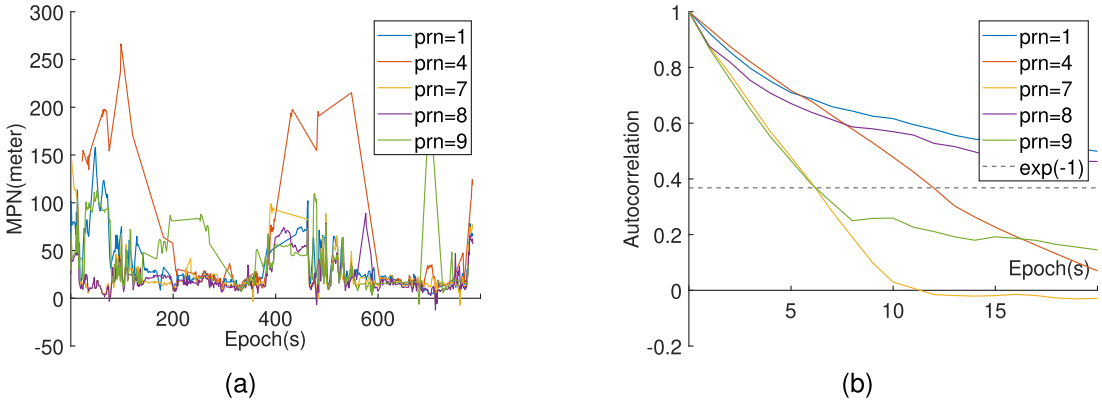

Fig. 12. MPN with time in HK medium urban trip (a); MPN autocorrelation in HK medium urban trip (b).

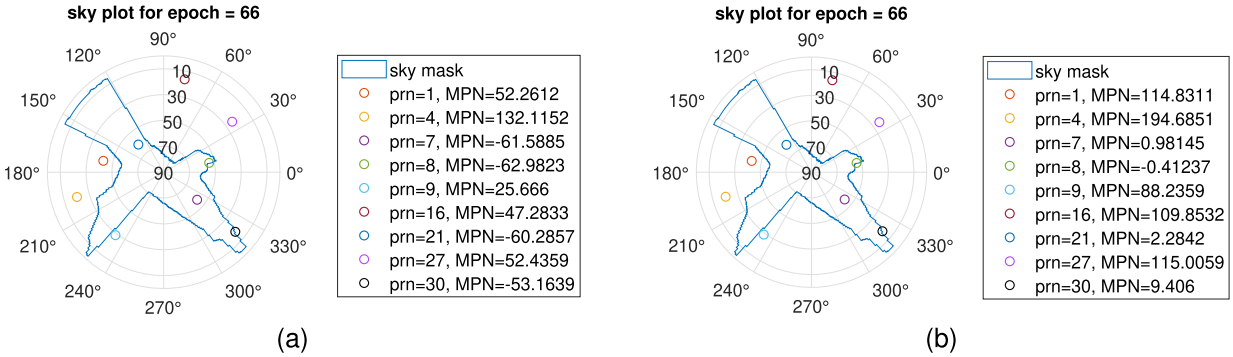

Fig. 13. Estimated MPN with a constant covariance matrix (left) and a varying covariance matrix (right) in the medium urban trip.

an autonomous driving system commonly equips up-to-date hardware and uses C++, Caffe on Linux, rather than MATLAB, which would improve the computational efficiency.

2) Various Feature Types Impact: To evaluate the impact of various feature types on MPN modeling, experiments were conducted using the estimated MPN of GPS satellites as the ground truth and different types of features as input on the HK medium urban trip. Specifically, the expected probability is set as $q = 0.99$ to obtain the predicted quantile of MPN distribution. In the HK medium urban trip, MPNMLP and MPNLSTM with time-series lengths $\tau = 5, 10, 15$, are trained using different feature inputs: 1. CN0 (S); 2. elevation angle (EL); 3. CN0 and elevation angle (ELS) 4. GNSS features, including elevation angle, azimuth angle, CN0 (GNSS); 5. GNSS and point cloud features (G_L); 6. GNSS and image features (G_C); 7. all multi-sensor features (AMS). The purpose of this experiment is to analyze the impact of different features and the effect of time series length on the modeling performance.

Table IV presents the averaged $\mathscr{R}$ results with the medium urban trip using different features and neural networks. Comparing the results using S and EL features, CN0 has a stronger correlation with multipath than elevation angle for all network models as $\mathscr{R}$ results with CN0 input are more immense than those with elevation angle input. That can be explained by the fact a satellite with a high elevation angle does not necessarily imply that the received signals only include direct LOS signals without reflection and diffraction. For example, an NLOS signal can have a high elevation angle from satellite A, while a LOS signal can have a low elevation angle from satellite B at the same epoch. However, the reflection and diffraction of signals do affect signal strength. Furthermore, the $\mathscr{R}$ results with feature ELS are better than single feature inputs of El and S, and the $\mathscr{R}$ results with GNSS are better than ELS, suggesting that elevation angle, CN0, and azimuth angle are all correlated with MPN. Additionally, point cloud features show a strong correlation with multipath comparing the results for feature input G_L and GNSS with over 5% improvement. Camera data is also helpful for modeling multipath errors by comparing the $\mathscr{R}$ results between G_C and GNSS. It is observed that G_L obtains better results than AMS for some cases. One possible explanation is that the AMS could have excessive features, making it challenging for the neural network to converge to the minimum. The front camera may contain information irrelevant to modeling. Despite this, the MPNLSTM_15 with AMS input has the best result. Therefore, AMS is concluded to be a suitable choice for feature inputs.

Table IV also shows that MPNLSTM has markable improvements in effectiveness compared with the benchmark model (MPNMLP) regardless of the feature inputs. For instance, the results increase from 0.6345 with MPNMLP to 0.8313 with MPNLSTM_15 for AMS input. Also, the result indicates that a longer time series length leads to better performance. As seen in results obtained with S feature input, the $\mathscr{R}$ values for time-series lengths of $\tau = 5, 10, 15$ are 0.4567, 0.5434 and 0.5728, respectively. The improvement from 5 to 10 is more evident than from 10 to 15, suggesting that the improvement declines as the length increases over a value, especially for S and EL feature inputs. This is consistent with the fact that the time constant equals 13.54 seconds. However, using MPNLSTM_05 with S or EL inputs does not significantly improve compared to the MPNMLP model with around 0.03 improvement. It is therefore important to choose a

TABLE III

HARDWARE AND SOFTWARE SPECIFICATION

CPU	GPU	Operating System	Machine Learning Platform	Programming Language
Intel i7-9700k	Nvidia RTX 2080	Windows 10	Matlab R2022a	Matlab

TABLE IV

$\mathscr{R}$ VALUE RESULTS FOR DIFFERENT INPUT AND DIFFERENT NETWORKS IN MEDIUM URBAN TRIP

	Features / Network	S	EL	ELS	GNSS	G_L	G_C	AMS
Medium Urban Trip	MPNMLP	0.4230	0.3078	0.4657	0.5525	0.6523	0.5992	0.6345
	MPNLSTM_05(5sec)	0.4567	0.3348	0.5164	0.5810	0.6876	0.6544	0.6696
	MPNLSTM_10(10sec)	0.5434	0.4211	0.5723	0.6254	0.7072	0.7035	0.7478
	MPNLSTM_15(15sec)	0.5728	0.4321	0.6102	0.6812	0.7410	0.7589	0.8313

TABLE V

THE PROCESSING SPEED (IN HZ) OF ONBOARD PART FOR REAL-TIME PROCESSING

Speed (Hz)	Features / Network	S	EL	ELS	GNSS	G_L	G_C	AMS
Medium Urban Trip	MPNMLP	40981	63376	51710	62470	2268	1000	1147
	MPNLSTM_05(5sec)	22266	19628	21865	21630	182	294	230
	MPNLSTM_10(10sec)	21638	21640	21993	22211	286	370	222
	MPNLSTM_15(15sec)	21673	23013	20822	22358	231	270	201

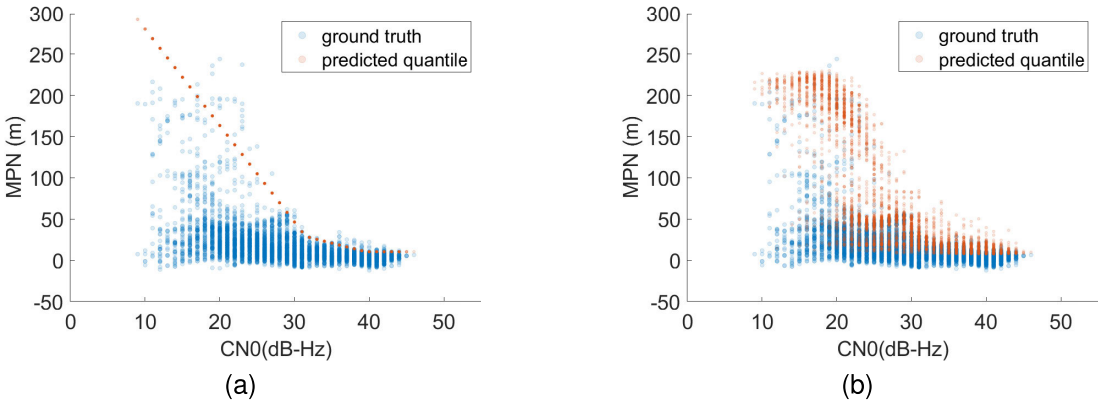

Fig. 14. Predicted results using MPNMLP (left) and MPNLSTM_05 (right) using single feature CN0 input in HK deep urban trip.

suitable length for modeling. In this study, a time-series length of 15 seconds is suggested.

Table V presents the processing speed of the onboard component for real-time processing in the medium urban trip. As the number of features increases, the processing speed decreases. For a vehicle speed of 40km/h in a metropolitan area, the slowest processing speed of 201Hz corresponds to a movement of 5 centimeters.

3) Comparison Effectiveness Between Designed Model and Benchmark: In this experiment, the purpose is to compare the effectiveness of the proposed LSTM model with the benchmark, MLP model, in [22] in different urban scenarios. Due to the necessity, only the top two input levels in Section III-C2, G_L and AMS, are selected for experiments, and the benchmark MPNMLP and the best performance model MPNLSTM_15 in the medium urban trip are tested in the HK deep and harsh urban trips to examine the performance of the LSTM and benchmark on different trips. The expected probability is still set as $q = 0.99$.

From Table VI, although the three trips have different MPN distributions leading to different $\mathscr{R}$ values, the MPNLSTM_15 results are always greater than those with MPNMLP. By using AMS as input, the effectiveness improves by 31%, 14.5%, and 16.8% for medium, deep and harsh urban scenarios, respectively. This consistent outperformance of the LSTM model over the previous MLP model highlights the significance of considering temporal correlation when modeling multipath errors in an urban environment. Furthermore, it is worth noting that the HK harsh urban trip demonstrates lower effectiveness compared to the other trips, indicating that modeling this particular trip is more challenging. This finding aligns with reality as the HK harsh urban trip involves a more complex surrounding scenario. It is important to note that the selected features in [22] have less information compared to the features used in this article. Consequently, the performance of the model in [22] may decrease when compared to the benchmark results presented here.

TABLE VI

$\mathscr{R}$ VALUE RESULTS FOR DIFFERENT TRIPS

Trips	Network	G_L	AMS
Medium Urban Trip	MPNMLP	0.6523	0.6345
	MPNLSTM_15	0.7410	0.8313
Deep Urban Trip	MPNMLP	0.6889	0.6994
	MPNLSTM_15	0.7576	0.7919
Harsh Urban Trip	MPNMLP	0.5062	0.5997
	MPNLSTM_15	0.6986	0.7007

Figures 14 and 15 provide a visual and intuitive comparison of the performance of MPNLSTM with MPNMLP in the deep and harsh trips, respectively, using S and EL input separately.

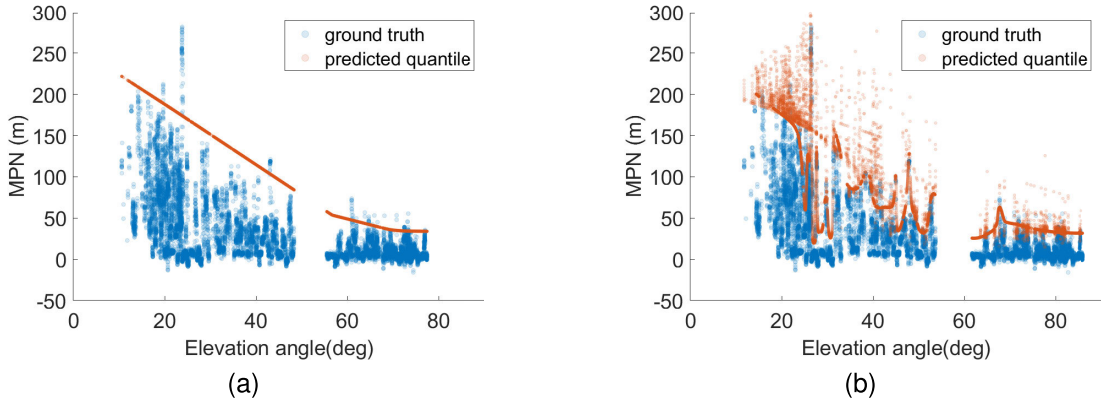

Fig. 15. Predicted results using MPNMLP (left) and MPNLSTM_15 (right) using single feature elevation angle input in HK harsh urban trip.

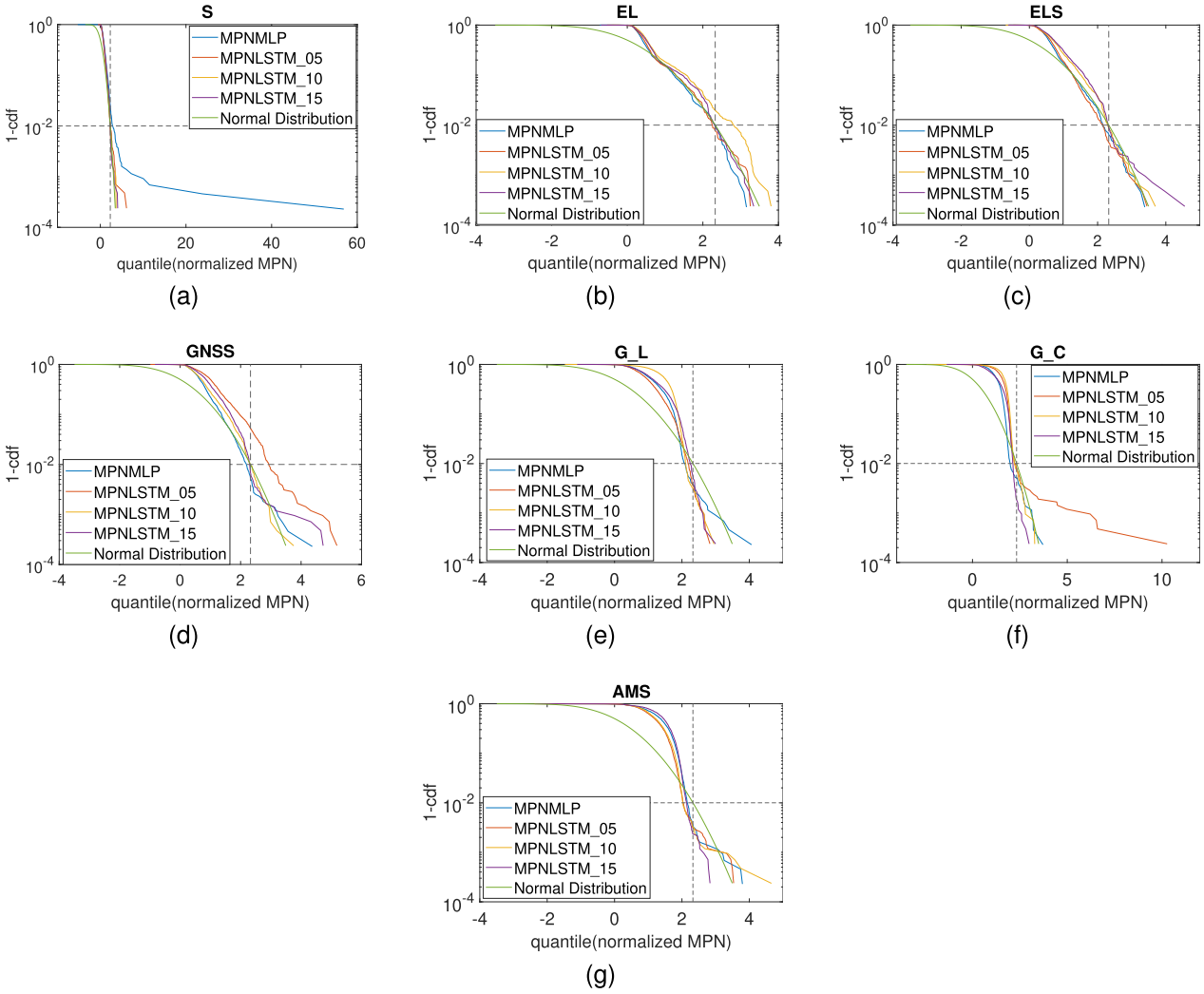

Fig. 16. plots of CDFs for four models and a Gaussian distribution with different inputs in the medium urban trip. (a) S. (b) EL. (c) ELS. (d) GNSS. (e) G_L. (f) G_C. (g) AMS.

The red dots on the y-axis represent the quantile value of the predicted MPN distribution, while the blue dots on the y-axis represent the ground truth.

The results in figure 14 and 15 reveal that MPNLSTM fits the data more precisely than MPNMLP, capturing a greater degree of distributional details. It should be noted that the MPNLSTM has multiple predicted values for each feature value since the output from MPNLSTM is determined by both the current epoch features and the features of previous epochs. Even if the features at the single epoch are the same, different predicted values can be obtained by MPNLSTM with varying features of earlier epochs. This visual representation reinforces

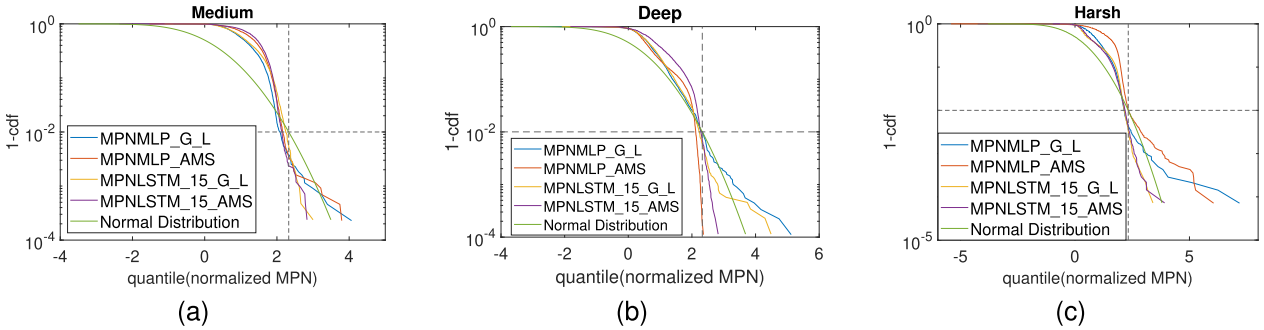

Fig. 17. plots of CDFs for three trips on MPNMLP and MPNLSTM model with G_L and AMS feature inputs. (a) Medium urban trip. (b) Deep urban trip. (c) Harsh urban trip.

the notion that the LSTM model is more effective in capturing the temporal correlation and accurately modeling the multipath errors in these challenging scenarios.

4) Conservatism Validation: This part is to validate the conservatism of the above models. The models' conservatism is validated by plotting the CDFs. Figures 16 and 17 show CDF plots from different models and feature inputs for the three trips. The x-axis represents the normalized multipath errors; the y-axis is $1-CDF(x)$. When the y-axis is $1-q$ and the model's quantile value is smaller than the quantile of the standard Gaussian distribution, the model is deemed to bound the MPN. In figure 16, the CDFs are illustrated with four different models, including MPNMLP (blue), MPNLSTM_05 (red), MPNLSTM_10 (yellow) and MPNLSTM_15 (purple), with varying inputs of feature in medium urban trips. The green line is the standard Gaussian distribution. Figure 17 shows the results for medium, deep, and harsh urban trips with G_L and AMS feature inputs, respectively.

Figures 16 and 17 demonstrate that all time-series models can be bounded in the three trips with G_L and AMS feature input. In contrast, the MPNMLP model can only overbound multipath errors with G_L and AMS feature input in the medium urban trip, failing to overbound in the deep urban trip with G_L feature input and in the harsh urban trip with AMS feature input. Therefore, time-series models exhibit greater capability in bounding multipath errors than the MPNMLP model. Furthermore, the results in figure 17 indicate that the MPNLSTM_15 can be bound between the standard Gaussian distribution line and the quantile line, which indicates that MPNLSTM_15 is better suited for bounding the tail probability of MPN after the quantile.

Comparing the results with different feature inputs in figure 16, LiDAR and camera data also play essential roles in overbounding, enabling all models to successfully overbound multipath errors. Therefore, this study suggests using AMS as a suitable feature input for a time-series model considering both the effectiveness and conservatism.

IV. Conclusion

A multipath overbounding method under urban environments is proposed based on long short-term memory (LSTM) to account for the temporal correlation. To further improve the performance and shorten the convergence time of the LSTM network, an optimized quantile loss function is designed. Experiments are conducted using urban data in Hong Kong with feature inputs from LiDAR, camera, and global navigation satellite systems (GNSS). The experiment results have demonstrated that LSTM is more suitable for modeling multipath errors than the benchmark multi-layer perceptron (MLP) with improved effectiveness and conservatism. Specifically, the LSTM network has higher $\mathscr{R}$ values than the MLP method regardless of any feature input based on the test results. The LSTM network with 15 seconds of time series length using the feature input from LiDAR, camera, and GNSS is the most effective model with 0.8313 $\mathscr{R}$ value in the medium urban trip. Also, the LSTM network can bound the multipath errors in medium, deep and harsh urban trips conservatively with the concatenation of GNSS, image and point cloud features, while MLP can only conservatively bounds medium and deep urban trips. Also, it is illustrated that features from LiDAR and camera data are critical for multipath modeling. To ensure the conservatism of the modeled multipath, this paper suggests implementing an LSTM model using 15 seconds of sequential GNSS, image, and point cloud features. The contributions of this work are threefold. Firstly, the proposed method for overbounding multipath errors considers the temporal correlation of multipath errors, resulting in better performance than the previous study. Secondly, this study is the first to employ LiDAR and camera data for overbounding multipath errors. Lastly, a novel quantile loss function is introduced, which is more efficient for training the LSTM network than the quantile regression loss and can be readily applied to other works.

Future work will focus on exploring the trade-off between computational time and model complexity to facilitate practical application. The fine-tuned network weights of the pre-trained backbones may be utilized to extract more relevant features, and the optimal time-series length for modeling multipath errors will also be investigated. Additionally, multi-sensor fusion methods will be deployed to enhance the model's performance.

Appendix
The Formal Derivation of k

The appendix presents a formal derivation of the zooming factor k, which ensures the theoretical convergence of the Huber quantile loss function to the desired quantile. Assuming a dataset $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ containing N samples drawn from an unknown distribution, a model of the relationship between x_i and y_i is expressed as follows:

$$y_i = \theta(x_i) + \epsilon \tag{19}$$

where $\theta(x)$ is a function to map the variable x to the space of variable y, ϵ is random noise from a known distribution. The goal of maximum likelihood estimation is to find a θ to maximize the joint probability, expressed as,

$$\hat{\theta} = \arg\max_{\theta}(\prod_{i=1}^{N} p(y_i|x_i,\theta)) = \arg\min_{\theta}(-\sum_{i=1}^{N} log(p(y_i|x_i,\theta))) \tag{20}$$

Also, the minimization of the QL function is expressed as,

$$\hat{\theta} = \arg\min_{\theta}(\mathcal{J}_q^{QL}(y,\theta(x))) = \arg\min_{\theta}(\sum_{i=1}^{N} \mathcal{L}_q^{QL}(y_i - \theta(x_i))) \tag{21}$$

Thus, minimizing the QL function is equivalent to the maximum likelihood estimate of θ if and only if

$$p(y_i|x_i,\theta) \propto e^{-\mathcal{L}_q^{QL}(y_i-\theta(x_i))} \tag{22}$$

And we assume that $f(y)$ is the PDF of $p(y_i|x_i,\theta)$, $F(y)$ is the CDF of $p(y_i|x_i,\theta)$ and $G(y)$ is the integral of $F(y)$. Based on (22) and $F(-\infty) = 1$, the $f(y)$, $F(y)$ and $G(y)$ can be expressed as,

$$f(y) = \begin{cases} (1-q)qe^{(1-q)(y-a)}, & y < a \\ (1-q)qe^{-q(y-a)}, & y \le a \end{cases} \tag{23}$$

$$F(y) = \begin{cases} qe^{(1-q)(y-a)}, & y < a \\ 1-(1-q)qe^{-q(y-a)}, & y \le a \end{cases} \tag{24}$$

$$G(y) = \begin{cases} \frac{q}{1-q}e^{(1-q)(y-a)}, & y < a \\ \frac{1}{1-q} - 1/q + \frac{1-q}{q}e^{-q(y-a)} + (y-a), & y \le a \end{cases} \tag{25}$$

where $a = \theta(x)$.

In the following derivation, Lemma 1 will be utilized. This basic result can be obtained based on the knowledge of indefinite integration, and its proof is omitted here.

Lemma 1: The indefinite integration is illustrated as $\int yf(y)dy = yF(y) - G(y) + C$, *where C is a constant.*

In order to ensure that minimizing the HQL function is equivalent to minimizing the QL function, an adjusted probability q_k needs to be derived based on q and σ. Specifically, the Huber quantile loss can be expressed as:

$$\begin{aligned} &\mathcal{J}_q^{HQL}(y,\theta(x)) \\ &= E(\mathcal{L}_q^{HQL}(y-a)) = \int_{-\infty}^{\infty} \mathcal{L}_q^{HQL}(y-a)f(y)dy \\ &= \int_{-\infty}^{a-\sigma} (1-q_k)(a-y-\frac{1}{2}\sigma)dF(y) \\ &\quad + \int_{a-\sigma}^{a} \frac{1-q_k}{2\sigma}(y-a)^2 dF(y) \\ &\quad + \int_{a}^{a+\sigma} \frac{q_k}{2\sigma}(y-a)^2 dF(y) \\ &\quad + \int_{a+\sigma}^{\infty} q_k(y-a-\frac{1}{2}\sigma)dF(y) \end{aligned} \tag{26}$$

To minimize the convex loss function, $\frac{\partial \mathcal{J}_q^{HQL}(y,a)}{\partial a}$ should equal to 0. By combining (26) with Lemma 1, it can be derived that

$$\begin{aligned} &\frac{\partial \mathcal{J}_q^{HQL}(y,a)}{\partial a} \\ &= \int_{-\infty}^{a-\sigma}(1-q_k)dF(y) + \int_{a-\sigma}^{a}\frac{1-q_k}{\sigma}(a-y)dF(y) \\ &\quad + \int_{a}^{a+\sigma}\frac{q_k}{\sigma}(a-y)dF(y) + \int_{a+\sigma}^{+\infty} -q_k dF(y) \\ &= (1-q_k)F(y)|_{-\infty}^{a-\sigma} + \frac{1-q_k}{\sigma}aF(y)|_{a-\sigma}^{a} \\ &\quad - \frac{1-q_k}{\sigma}(yF(y)-G(y))|_{a-\sigma}^{a} + \frac{q_k a}{\sigma}F(y)|_{a}^{a+\sigma} \\ &\quad - \frac{q_k}{\sigma}(yF(y)-G(y))|_{a}^{a+\sigma} - q_k F(y)|_{a+\sigma}^{+\infty} \\ &= -G(a-\sigma)\frac{1-q_k}{\sigma} + G(a)(\frac{1-q_k}{\sigma} - \frac{q_k}{\sigma}) \\ &\quad + G(a+\sigma)\frac{q_k}{\sigma} - q_k = 0 \end{aligned} \tag{27}$$

From (25), $G(a) = \frac{q}{1-q}$, $G(a-\sigma) = \frac{q}{1-q}e^{-(1-q)\sigma}$, and $G(a+\sigma) = \frac{1}{1-q} - \frac{1}{q} + \frac{1-q}{q}e^{-q\sigma} + \sigma$ can be obtained. Substituting the above three formulas into (27) yields:

$$\begin{aligned} &\frac{\partial \mathcal{J}_q^{HQL}(y,a)}{\partial a} \\ &= -\frac{1-q_k}{\sigma}\frac{q}{1-q}e^{-(1-q)\sigma} + \frac{q}{1-q}(\frac{1-q_k}{\sigma} - \frac{q_k}{\sigma}) \\ &\quad + \frac{q_k}{\sigma}(\frac{1}{1-q} - \frac{1}{q}) + \frac{q_k}{\sigma}\frac{1-q}{q}e^{-q\sigma} + \frac{q_k}{\sigma}\sigma - q_k = 0 \end{aligned} \tag{28}$$

q_k can be derived from (28), as follows:

$$q_k = \frac{q^2(1-e^{-(1-q)\sigma})}{q^2(1-e^{-(1-q)\sigma}) + (1-q)^2(1-e^{-q\sigma})} \tag{29}$$

Therefore, the zooming factor k can be derived by dividing both sides of the (29) by q, as follows:

$$k = \frac{q(1-e^{-(1-q)\sigma})}{q^2(1-e^{-(1-q)\sigma}) + (1-q)^2(1-e^{-q\sigma})} \tag{30}$$

□

ACKNOWLEDGMENT

The authors extend their gratitude to the creators of the UrbanNav dataset for their generosity in making the dataset openly available for research purposes.

REFERENCES

[1] S. Gleason, D. Gebre-Egziabher, and D. G. Egziabher, *GNSS Applications and Methods*. Norwood, MA, USA: Artech House, 2009.

[2] M. S. Grewal, A. P. Andrews, and C. G. Bartone, *Global Navigation Satellite Systems, Inertial Navigation, and Integration*. Hoboken, NJ, USA: Wiley, 2020.

[3] Y. Jiang and J. Wang, "A new approach to calculate the horizontal protection level," *J. Navigat.*, vol. 69, no. 1, pp. 57–74, Jan. 2016.

[4] H. Jing, Y. Gao, S. Shahbeigi, and M. Dianati, "Integrity monitoring of GNSS/INS based positioning systems for autonomous vehicles: State-of-the-art and open challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 14166–14187, Sep. 2022.

[5] T. G. R. Reid et al., "Localization requirements for autonomous vehicles," 2019, *arXiv:1906.01061*.

[6] S. Bijjahalli and R. Sabatini, "A high-integrity and low-cost navigation system for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 356–369, Jan. 2021.

[7] J. Rife and B. Pervan, "Overbounding revisited: Discrete error-distribution modeling for safety-critical GPS navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 2, pp. 1537–1551, Apr. 2012.

[8] M. Harris, P. Schlais, T. Murphy, A. Joseph, and J. Kazmierczak, "GPS and Galileo airframe multipath error bounding method and test results," in *Proc. 33rd Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2020, pp. 114–139.

[9] B. DeCleene, "Defining pseudorange integrity-overbounding," in *Proc. 13th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Sep. 2000, pp. 1916–1924.

[10] C. A. Shively and R. Braff, "An overbound concept for pseudorange error from the LAAS ground facility," in *Proc. IAIN World Congr. 56th Annu. Meeting Inst. Navigat.*, 2000, pp. 661–671.

[11] S. Khanafseh, S. Langel, and B. Pervan, "Overbounding position errors in the presence of carrier phase multipath error model uncertainty," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2010, pp. 575–584.

[12] M. Felux, M.-S. Circiu, S. Caizzone, C. Enneking, F. Fohlmeister, and M. Rippl, "Towards airborne multipath models for dual constellation and dual-frequency GNSS," in *Proc. Int. Tech. Meeting The Inst. Navigat.*, Feb. 2019, pp. 62–68.

[13] Z. Gao, K. Fang, Z. Wang, K. Guo, and Y. Liu, "An error overbounding method based on a Gaussian mixture model with uncertainty estimation for a dual-frequency ground-based augmentation system," *Remote Sens.*, vol. 14, no. 5, p. 1111, Feb. 2022.

[14] X. Fang, D. Song, C. Shi, L. Fan, and Z. Hu, "Multipath error modeling methodology for GNSS integrity monitoring using a global optimization strategy," *Remote Sens.*, vol. 14, no. 9, p. 2130, Apr. 2022.

[15] N. Alam and A. G. Dempster, "Cooperative positioning for vehicular networks: Facts and future," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1708–1717, Dec. 2013.

[16] G. De Angelis, G. Baruffa, and S. Cacopardi, "GNSS/cellular hybrid positioning system for mobile users in urban scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 1, pp. 313–321, Mar. 2013.

[17] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Feb. 2012.

[18] J. Rife, S. Pullen, B. Pervan, and P. Enge, "Paired overbounding and application to GPS augmentation," in *Proc. PLANS. Position Location Navigat. Symp.*, Jul. 2004, pp. 439–446.

[19] J. Blanch, T. Walter, and P. Enge, "A MATLAB toolset to determine strict Gaussian bounding distributions of a sample distribution," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Nov. 2017, pp. 4236–4247.

[20] S. Khanafseh et al., "GNSS multipath error modeling for automotive applications," in *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2018, pp. 1573–1589.

[21] G. De Angelis, A. Moschitta, and P. Carbone, "Positioning techniques in indoor environments based on stochastic modeling of UWB round-trip-time measurements," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 8, pp. 2272–2281, Aug. 2016.

[22] H. No and C. Milner, "Machine learning based overbound modeling of multipath error for safety critical urban environment," in *Proc. 34th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2021, pp. 180–194.

[23] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Math. Control, Signals, Syst.*, vol. 2, no. 4, pp. 303–314, Dec. 1989.

[24] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation With Applications to Tracking and Navigation: Theory Algorithms and Software*. Hoboken, NJ, USA: Wiley, 2001.

[25] C. Amielh, A. Chabory, C. Macabiau, and L. Azoulai, "Validation of existing GNSS multipath model," in *Proc. 30th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Nov. 2017, pp. 1772–1789.

[26] N. Kubo, K. Kobayashi, and R. Furukawa, "GNSS multipath detection using continuous time-series C/N0," *Sensors*, vol. 20, no. 14, p. 4059, Jul. 2020.

[27] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Stat.*, vol. 35, no. 1, pp. 73–101, 1964, doi: 10.1214/aoms/1177703732.

[28] T. Ozeki and N. Kubo, "A new approach of detecting NLOS signals based on modified residual error check," in *Proc. 34th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2021, pp. 138–167.

[29] L. Icking, G. Zhang, L.-T. Hsu, and S. Schön, "Quantification of GNSS NLOS spatial correlation—A case study in Hong Kong's urban Canyon," in *Proc. Int. Tech. Meeting The Inst. Navigat.*, Feb. 2022, pp. 712–722.

[30] M. Felux, M.-S. Circiu, S. Caizzone, M. Rippl, M. Sgammini, and P. Durel, "Standardization of new airborne multipath models," in *Proc. Int. Tech. Meeting Inst. Navigat.*, Feb. 2020, pp. 146–153.

[31] H.-F. Ng, G. Zhang, and L.-T. Hsu, "A computation effective range-based 3D mapping aided GNSS with NLOS correction method," *J. Navigat.*, vol. 73, no. 6, pp. 1202–1222, Nov. 2020.

[32] T. Suzuki and N. Kubo, "Simulation of GNSS satellite availability in urban environments using Google Earth," in *Proc. Pacific PNT Meeting*, 2015, pp. 1069–1079.

[33] J. Wang, "Analysis and design of a recurrent neural network for linear programming," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 9, pp. 613–618, Sep. 1993.

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural Comput.*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000.

[35] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, 2005.

[36] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[37] K. Smagulova and A. P. James, "A survey on LSTM memristive neural network architectures and applications," *Eur. Phys. J. Special Topics*, vol. 228, pp. 2313–2324, Oct. 2019.

[38] M. Irsigler, J. A. Avila-Rodriguez, and G. W. Hein, "Criteria for GNSS multipath performance assessment," in *Proc. 18th Int. Tech. Meeting Satell. Division Inst. Navigat.*, 2005, pp. 2166–2177.

[39] L. J. Greenstein, V. Erceg, Y. S. Yeh, and M. V. Clark, "A new path-gain/delay-spread propagation model for digital cellular channels," *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 477–485, May 1997.

[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5099–5108.

[42] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, no. 1, pp. 33–50, Jan. 1978.

[43] L.-T. Hsu et al., "UrbanNav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas," in *Proc. 34th Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2021, pp. 226–256.

Ruirui Liu received the B.Eng. degree from the Department of Electronic Information Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2016, and the M.Eng. degree from the Department of Electrical Engineering, University of Michigan, USA, in 2018. He is currently pursuing the Ph.D. degree with The Hong Kong Polytechnic University. His research interests include autonomous driving localization's integrity and deep learning on navigation.

Yiping Jiang received the Ph.D. degree from the University of New South Wales, Sydney, Australia, in 2014. She is currently an Assistant Professor with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. She lectures on avionics system and satellite navigation for undergraduate and postgraduate students. Her research interests include precise positioning and integrity monitoring for civil aviation and intelligent transportation systems.