

# Research Article

# **T-SPP: Improving GNSS Single-Point Positioning Performance Using Transformer-Based Correction**

Fan Wu<sup>(b)</sup>,<sup>1</sup> Liangrui Wei<sup>(b)</sup>,<sup>1</sup> Haiyong Luo<sup>(b)</sup>,<sup>2</sup> Fang Zhao<sup>(b)</sup>,<sup>1</sup> Xin Ma<sup>(b)</sup>,<sup>1</sup> and Bokun Ning<sup>(b)</sup>

<sup>1</sup>School of Software Engineering, Beijing University of Posts and Telecommunications, Beijing, China <sup>2</sup>Research Center for Ubiquitous Computing Systems, Institute of Computing Technology Chinese Academy of Sciences, Beijing, China

Correspondence should be addressed to Haiyong Luo; yhluo@ict.ac.cn

Received 14 June 2023; Revised 18 January 2024; Accepted 9 February 2024; Published 29 February 2024

Academic Editor: Vasudevan Rajamohan

Copyright © 2024 Fan Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

GNSS (global navigation satellite systems) technology enables high-precision single-point positioning (SPP) in open environments. However, the accuracy of GNSS positioning is significantly compromised in complex urban canyons due to signal obstructions and nonline-of-sight propagation errors. To address this challenge, we propose a GNSS displacement estimation algorithm. This method learns nonlinear dependencies between GNSS raw measurements and corresponding position changes, capturing dynamic and layered features in GNSS measurement data for displacement estimation. We introduce a denoising auto-encoder (DAE) to preprocess raw GNSS observations, reducing the impact of noise. The model simultaneously outputs estimated displacement and model confidence. The fusion process dynamically combines positioning results from the SPP algorithm and the D-Tran model, adaptively blending them to achieve accurate and optimal positioning estimation. This approach optimizes the accuracy of estimated positioning results while maintaining confidence in the estimation. Experimental results show a 61% reduction in root mean square error (RMSE) and 100% availability in urban canyon environments compared to traditional single-point positioning techniques.

# 1. Introduction

Nowadays, demand for high-accuracy position-based services in daily activities, such as location-based services (LBS) [1] and intelligent transportation system (ITS) [2], has increased significantly. This has generated interest in improving the localization accuracy of low-cost receivers, specifically those integrated into smartphones, within the GNSS (global navigation satellite systems) [3]. In GNSS positioning with smartphones, accurate absolute positioning extraction is essential, particularly in complex environments. Smartphones typically use a combination of GNSS code-only SPP (singlepoint positioning) and IMU (inertial management unit) data to derive location information. While SPP performance is satisfactory in areas with good satellite observing conditions like rural, expressway, and suburban areas, realistic usage of SPP in semiurban, forested, and urban areas presents numerous limitations and challenges, including poor multipath

suppression, low carrier-to-noise density ratio (C/N0), and missing measurements, as exemplified in Figure 1. The notorious multipath and NLOS (non-line-of-sight) effects in dense building and scenario environments significantly reduce SPP localization accuracy and continuity [4]. In high multipath environments and kinematic scenarios, the SPP solution can degrade to several tens-of-meters. The extraction of stable and continuous positioning information is an essential requirement for various everyday applications. Hence, a significant number of academic researchers aim to improve positioning performance.

Numerous studies have thoroughly investigated various methods to improve the accuracy and continuity of global navigation satellite system (GNSS) positioning. These methods can typically be classified into two categories: external hardware and error reduction. In recent years, there has been considerable research interest in integrating GNSS with other sensors. For instance, studies have proposed



FIGURE 1: The urban positioning environment of SPP with NLOS and multipath.

integrating vision sensors [5, 6], lidar odometers [7, 8], and INS (inertial navigation systems) [9] with GNSS to improve positioning accuracy in environments where GNSS signals are degraded. While methods relying on IMU can enhance positioning accuracy, there are notable drawbacks. Firstly, the increased use of sensors introduces additional costs. Secondly, such methods may be sensitive to sensor configuration, requiring precise installation and calibration to ensure accurate attitude and motion estimation. Lastly, IMU and similar sensors can only provide relative positions, and achieving precise carrier positioning still relies on sensors primarily based on GNSS to provide the absolute position of the carrier. Nonetheless, GNSS still remains the primary sensor for providing absolute user position, which necessitates high levels of accuracy. Since 2016, carrier-phase and pseudorange measurements have been made publicly available through Google's Android API (application programming interface) [10]. This has paved the way for researchers to analyze and condition raw measurements from smartphones, such as carrier-phase smoothing and Doppler smoothing, to improve positioning accuracy [11]. Simultaneously, predicting interferences in urban areas [12] or measurement noise [13] has provided novel approaches for enhancing positioning performance. It is worth noting that non-Gaussian error distributions in practical global navigation satellite system (GNSS) measurements, as well as environmental interferences, can lead to degradation in the accuracy of traditional techniques that rely on Gaussian approximations for error estimation [14].

Over the past decade, the rapid advancement of deep learning techniques has enabled the utilization of deep learning models to acquire abstract representations of preexisting positioning data, which can be utilized to direct the navigation system's behavior when satellite signals are weak or when the number of visible satellites is less than four. In order to enhance positioning accuracy, researchers have explored the fusion of deep learning models with integrated navigation methods [15, 16]. Nonetheless, these methods rely on the integration of intelligent techniques with INS, such as the method proposed by Fang et al. [17], who used LSTM (long short-term memory) to predict pseudomeasurement data. These methods relying on deep kinematic models first struggle to effectively handle measurement errors and uncertainties, especially in complex urban environments, and cannot avoid the accumulation of errors during the computation process. Additionally, deep kinematic models still depend on inertial navigation systems (INS), requiring precise installation and calibration of sensors. This increases the complexity of the system.

Different from the methods relying on IMU and deep kinematic modeling as mentioned above, this paper proposes a GNSS measurement feature learning framework based on the transformer model to enhance the continuity and accuracy of position estimation through an intelligent SPP algorithm. In contrast to the LSTM model used in the approach proposed by paper [17], we employ a parallelizable transformer model. This choice considers the limitations of the LSTM model, such as gradient vanishing or explosion during training, lengthy training time, and limited exploration of long-term dependencies in the data. The transformer model enhances the training speed and better explores long-term dependencies in the data. To enhance the model's robustness, we introduce a DAE (denoising autoencoder) model to clean GNSS raw observation data. The DAE model processes noise and outliers in the input data and reconstructs the input data, thereby improving the accuracy and robustness of the model. In summary, our proposed framework combines the strengths of deep learning and GNSS to provide a more accurate and robust approach for GNSS-based position estimation. In the first place, D-Trans introduces the capability to dynamically and hierarchically learn features from GNSS measurement data. This implies that the model can adapt to different environments and dynamic conditions, enhancing its adaptability in complex terrains such as urban canyons. Additionally, GNSS raw observations exhibit significant noise, which can impact model training. This paper introduces a denoising auto-encoder (DAE) to preprocess raw GNSS observations, aiding in mitigating the influence of noise on the model. This enhances the model's availability to measurement errors caused by signal obstruction and nonline-of-sight propagation in the environment.

The main contributions of this paper are summarized as follows:

- (i) This paper proposes a new model, D-Tran (denoising auto-encoder-transformer), which captures dynamic and potential features in GNSS measurement sequences. D-Tran is based on the transformer framework and uses an attention mechanism to capture global dependencies between GNSS measurements and position increments ( $\Delta P$ ). Additionally, a DAE is connected with the transformer to reduce occasional noise in GNSS measurements and produce denoised feature vectors. To the best of our knowledge, this work is the first to explore the application of deep learning methods only using the raw GNSS observation and has shown promising results in improving positioning performance.
- (ii) We propose a dynamic confidence estimation method that adaptively adjusts the weights of the intelligent model. This method allows for the adaptive and real-time adjustment of model weights, enabling the system to effectively adapt to environmental changes and optimize positioning performance. To address the challenge of lacking label for confidence estimation, we design a computation method for the loss function that incorporates confidence feedback. By incorporating confidence estimation into the loss function calculation, we are able to reflect the confidence information during training.
- (iii) We validated our proposed method by testing it on three datasets: the open-source UrbanNav dataset, our own device-collected dataset, and a proprietary dataset provided by Didi Corporation. Extensive experimental results show that our proposed T-SPP algorithm effectively enhances the accuracy, continuity, availability, and generalization performance of the SPP algorithm.

The remaining part of this paper is organized as follows: the relevant literature is reviewed in Section 2; Section 3 presents the proposed T-SPP framework for improving the continuity and performance of the SPP algorithm; in Section 4, we present the experimental results; and finally, the conclusion is given in Section 5.

## 2. Related Work

Achieving high-accuracy positioning in urban environments has been a focal point of previous studies. Prior to the availability of the Google API, GNSS chipsets only provided position, velocity, and time (PVT) information. Some studies attempted to mitigate positioning errors through loosely integrated navigation techniques [18, 19]. However, achieving high-accuracy positioning through loosely integrated navigation without relying on expensive external hardware and GNSS raw observations from smartphones is still a challenging task. This paper conducts an analysis of the current research progress from two perspectives: model-

based methods and data-driven modeling approaches. In May 2016, Google's announcement on the availability of raw GNSS measurements to the public brought about a significant breakthrough in the field of positioning performance. Since then, many studies have leveraged these raw observations to enhance positioning accuracy. For instance, Zhu et al. [20] proposed a method to separate the effects of multipath and NLOS by utilizing satellite elevation angle and C/N0, while Shuai et al. [21] introduced satellite broadcast data quality and user equivalent distance error for optimal satellite selection. To address cumulative errors from ionosphere, cycle slips, and outliers, a TT-SD (Three-Thresholds and Single-Difference) Hatch filter [11] was proposed, which adapts the carrier-phase smoothed pseudorange window width based on three thresholds. Prochniewicz et al. [22] investigated correlation models among different GNSS measurement values. For each signal and satellite block of the GPS, GLONASS, Galileo, and BeiDou systems, they established independent empirical stochastic models, considering crosscorrelation and temporal correlation among observation values. However, some errors do not behave as expected, and Farooq et al. [23] attempted to apply the extended Kalman filter (EKF) model to single-frequency pseudorange measurements. To accurately describe the user's movement, Guo et al. [24] utilized pseudorange and high-precision carrier phase observations to construct the state equation. Furthermore, a modified single-frequency precise point positioning (PPP) strategy was proposed by estimating separate clock biases for pseudorange and carrier-phase observations [25, 26]. However, in complex urban canyons, signal interruptions and ambiguity reinitialization often degrade the availability of PPP. Wen and Hsu [27] formulated a factor graph-based approach for GNSS positioning that effectively explores the time-correlation of GNSS measurements. Nevertheless, in complex urban environments, the trajectory can still deviate significantly from the ground truth trajectory. However, model-based GNSS positioning methods exhibit certain performance limitations in real-world scenarios. Due to the complexity of real environments, optimization methods for GNSS positioning based on model assumptions often deviate from these assumptions, leading to issues such as positioning divergence. Model-based approaches for mitigating GNSS single-point positioning errors suffer from poor fitting performance, limited generalization, and difficulty in adapting to the complexities of dynamic urban environments.

The field of artificial intelligence (AI) has experienced a rapid evolution, leading to a growing preference for datadriven approaches. In numerous studies, deep learning or machine learning techniques have been utilized to extract representations of GNSS observation data without requiring expertise in the underlying principles. Additionally, AI methods have been used to forecast complex GNSS measurement errors. For instance, Linty et al. [28] employed machine learning to identify amplitude ionospheric scintillation events, while Kaselimi et al. [29] combined a convolutional neural network and a gated recurrent unit to estimate ionospheric delays on GNSS satellite signals. Additional studies [30-33] have focused on the utilization of AI technology, including deep learning and machine learning, for automatic detection and classification of GNSS interference, such as jamming, spoofing, NLOS, and multipath signals. It is noteworthy that Munin et al. [34] employed a convolutional neural network (CNN) for the extraction of multipath detection features. Similarly, Min et al. [35] proposed a code multipath mitigation method using a deep neural network (DNN) for GNSS navigation, while Zhang et al. [15] employed a combination of LSTM and conventional fully connected neural networks (FCNNs) to predict satellite visibility and pseudorange uncertainty. However, it is important to note that these studies have primarily concentrated on the GNSS measurement domain rather than the GNSS positioning domain. While research based on GNSS observation data has controlled GNSS observation signals, ensuring high-quality GNSS observation data for calculating carrier positions, in complex environments, the available amount of observation data is limited. The data that have undergone filtering are even more challenging to meet the minimum requirements for satellite observation data in positioning, leading to the inability to output positioning results.

The primary objective of this study is to develop a robust and efficient GNSS positioning algorithm optimized for urban environments. To accomplish this, we propose a novel approach that transforms conventional GNSS positioning methods into a problem of estimating position increments using GNSS measurements. This approach employs an endto-end deep learning model in conjunction with a DAE to jointly predict and rectify GNSS positioning errors. Recognizing the limited interpretability of neural network methods, we assert that deep learning methods cannot fully replace the traditional SPP algorithm. Consequently, we integrate confidence estimation into the deep learning model to facilitate a better fusion of deep learning techniques with the traditional SPP algorithm. While previous studies, such as Kanhere et al. [36], have proposed algorithms for GNSS positioning corrections using DNNs, they did not address the issue of positioning continuity in urban environments, which is a key focus of our work. Our aim is to develop a highly effective and reliable GNSS positioning algorithm that can overcome the challenges posed by urban environments and enable continuous positioning for users.

#### 3. Proposed Method

In this section, we will discuss our methodology for developing the D-Tran model, which utilizes the raw GNSS measurements to estimate position increments. The overall architecture of our proposed T-SPP algorithm is illustrated in Figure 2. Figure 2 shows that the proposed T-SPP algorithm is built upon the transformer framework and consists of two main components. In the figure, we denote the predicted position increment as  $\Delta P_g$ ,  $\eta$  represents the estimation of model confidence,  $\Delta \hat{P}$  is the corrected and fused position increment, and  $\Delta P_T$  represents the true position increment. Other model parameters are described in detail in Section 3.2. The first component is the traditional SPP algorithm, which is represented by the blue rectangle. The GNSS OBS block contains the GNSS raw observation data, such as pseudorange and Doppler frequency. During the data preprocessing stage, gross errors are eliminated. Additionally, during the error correction phase, ionospheric errors, tropospheric errors, and other errors are corrected.

The D-Tran module, as the second component of the T-SPP method, employs a transformer to estimate position increments and model confidence by leveraging raw GNSS observation sequences. In situations where accurate labels are unavailable for determining the confidence of the intelligent model, a loss function is devised to integrate confidence estimation and position increment estimation, providing valuable feedback. When the number of observable satellites exceeds four, indicating the capability of the traditional SPP algorithm to yield positioning results, the algorithm dynamically merges the positioning results from the SPP algorithm with the position increments generated by the intelligent model, utilizing the predicted model confidence. However, in complex urban environments, the traditional SPP algorithm may fail to produce positioning results. To ensure continuity in positioning, the intelligent model is employed to deliver continuous positioning results in such scenarios.

Due to the lack of interpretability in deep learning models, they cannot completely replace the traditional SPP method for position estimation. On the contrary, deep learning methods should be seen as complementary to traditional SPP algorithms. These methods can optimize the positioning results obtained from SPP when it is capable of producing positioning results and provide additional positioning information when SPP fails to accurately determine the position.

3.1. SPP Observation Equation. In general, the pseudorange observation on a single frequency [37] between a satellite and a receiver can be modeled as shown in (1).

$$\rho_r^s = \mathbf{p}_r^s + c\delta t_r - c\delta t^s + I_r^s + T_r^s + \varepsilon_r^s, \tag{1}$$

where the superscript *s* indicates the number of satellite and the subscript *r* denotes the number of receiver;  $\rho_r^s$  is the measured pseudorange in meters;  $\mathbf{p}_r^s$  means the receiversatellite geometric distance in meters; *c* is the speed of light in vacuum in meters per second;  $\delta t_r$  and  $\delta t^s$  represent the receiver and satellite clock offset in seconds, respectively;  $I_r^s$ and  $T_r^s$  are ionospheric delay and tropospheric delay, respectively;  $\varepsilon_r^s$  includes unmodeled errors such as measurement noise and multipath error in meters.

In (1), the satellite clock offset can be computed using the broadcast ephemeris, which is transmitted from the GPS satellites and contains information about the satellite



FIGURE 2: The architecture of the T-SPP algorithm.

position and clock bias. To correct for the ionospheric and tropospheric delay, we use the Klobuchar model [38] and Saastamoinen model [39], respectively. The Klobuchar model is an empirical model based on the ionospheric electron density, while the Saastamoinen model accounts for the delay caused by the Earth's atmosphere. In the SPP algorithm part, we neglect the unmodeled errors, which include measurement noise and multipath error [40]. The distance between the satellite and the receiver can be calculated using the measured pseudorange, the speed of light in vacuum, and the receiver-satellite geometric distance. To obtain precise positioning results, a minimum of four observable satellites' raw data is necessary. This requirement arises from the presence of unknown variables, including three-dimensional position coordinates and receiver clock offset. The computation process involves solving multiple equations using the least squares method, and this can be expressed as the following equation:

$$\mathbf{p}_r^s + c\delta t_r = \rho_r^s + c\delta t^s - I_r^s - T_r^s - \varepsilon_r^s = \rho_r^{s} \hat{\rho_r}^s, \qquad (2)$$

$$\begin{cases} \sqrt{(x^{s_1} - x_r)^2 + (y^{s_1} - y_r)^2 + (z^{s_1} - z_r)^2} + c\delta t_r = \widehat{\rho_r^{s_1}}, \\ \sqrt{(x^{s_2} - x_r)^2 + (y^{s_2} - y_r)^2 + (z^{s_2} - z_r)^2} + c\delta t_r = \widehat{\rho_r^{s_2}}, \\ \dots \\ \sqrt{(x^{s_n} - x_r)^2 + (y^{s_n} - y_r)^2 + (z^{s_n} - z_r)^2} + c\delta t_r = \widehat{\rho_r^{s_n}}, \end{cases}$$
(3)

where  $(x^s, y^s, z^s)$  are the satellite coordinates in meters;  $(x_r, y_r, z_r)$  are the receiver coordinates in meters;  $\rho_{to}^{so}$  is the pseudorange after error correction. According to (3), users can obtain their position by receiving pseudorange observation from at least four satellites.

*3.2. D-Tran Model.* The detailed architecture of the D-Tran model is depicted in Figure 3. The inputs of our proposed D-Tran model are the raw GNSS measurements **S**, which are defined as follows:

$$\mathbf{S} = \{S_1, S_2, \dots S_n\},\tag{4}$$

$$S_{k} = \{p^{k}, \rho^{k}, f_{d_{k}}, c_{k}\},$$
(5)

where k represents the number of satellites,  $\rho_k$  is the satellite coordinates of satellite k,  $\rho_k$  is the pseudorange between satellite k and phone position,  $f_{d_k}$  is the Doppler

shift of satellite k, and  $c_k$  is the C/N0 of satellite k. To speed up training and improve the stability of the model, all the input data are normalized as shown in the following equation:

$$\mathbf{S} = \frac{\mathbf{S} - \mathbf{S}_{\min}}{\mathbf{S}_{\max} - \mathbf{S}_{\min}}.$$
 (6)

It is worth noting that raw GNSS measurement data are usually corrupted by noise, which can severely degrade the prediction accuracy of the model. Therefore, the proposed D-Tran model incorporates a DAE as a preprocessing step. The DAE is an unsupervised learning method that learns to extract useful features from the noisy input data. It consists of two stages, namely, the encoding stage and the decoding stage. In the encoding stage, noisy training samples  $\tilde{S}$  are mapped to a lower-dimensional space, and the denoised data are then reconstructed in the decoding stage, as shown in (7) and (8).



FIGURE 3: The overall architecture of the D-Tran model.

This preprocessing step effectively reduces the impact of noise on the subsequent processing steps and thus improves the overall performance of the model.

$$\boldsymbol{\eta} = \sigma (\mathbf{W} \cdot \tilde{\mathbf{S}} + \mathbf{b}), \tag{7}$$

$$\mathbf{X} = \sigma \left( \mathbf{W}' \cdot \mathbf{\eta} + \mathbf{b}' \right), \tag{8}$$

where **W** and **W**' are encoding and decoding weight matrices, respectively, **b** and **b**' are biases of input and output layers, and  $\sigma(\cdot)$  is the activation function defined as

$$\sigma(x) = \frac{1}{1 + \exp\left(-x\right)}.$$
(9)

The D-Tran model utilizes a denoising process through a DAE, where the resultant vector **X** is input into a transformer architecture to extract degradation features. The transformer is a sequence-to-sequence model with an encoder and decoder, designed to capture contextual information from neighboring positions [41]. The encoder predicts the continuous position increment from GNSS measurements in the proposed method. The transformer layer contains the multihead attention sublayer and the feed forward sublayer. The former aims to capture dependencies between input features, irrespective of their distance in the sequence. The input features X are encoded to generate queries' matrix Q and keys' matrix K, as depicted in (10). By performing such encodings, the transformer architecture effectively extracts features from the input data that capture the underlying degradation patterns, as demonstrated in this study.

$$\begin{cases} \mathbf{Q} = \mathbf{X} \times \mathbf{W}_{\mathbf{q}}, \\ \mathbf{K} = \mathbf{X} \times \mathbf{W}_{\mathbf{k}}, \end{cases}$$
(10)

where  $\mathbf{W}_q$  and  $\mathbf{W}_k$  are the learnable parameter matrices of different networks. The attention weight matrix  $\mathbf{A}$  can be formulated as (11).

$$\mathbf{A} = \operatorname{soft} \max\left(\frac{\mathbf{Q}\mathbf{K}^{T}}{\sqrt{d_{x}}}\right).$$
(11)

In the final step, the transformer performers feature encoding on input features to obtain the values V and the output is computed as follows:

$$\mathbf{V} = \mathbf{X} \times \mathbf{W}_{\mathrm{v}},\tag{12}$$

$$Attention (\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{A} \times \mathbf{V}, \tag{13}$$

where  $\mathbf{W}_{\nu}$  is the parameter matrix. The multihead attention is defined as follows:

Multihead (
$$\mathbf{Q}, \mathbf{K}, \mathbf{V}$$
) = Concat (head<sub>1</sub>, . . . head<sub>h</sub>) $\mathbf{W}_{O}$ ,  
head<sub>i</sub> = Attention ( $\mathbf{Q}_{i}, \mathbf{K}_{i}, \mathbf{V}_{i}$ ),  
(14)

where  $W_o$  is the learnable parameter matrix.

The feed forward sublayer, applied after the attention sublayer, is designed to further fine-tune the data dimension. This layer consists of two linear layers and a nonlinear activation function and can be denoted as the following function:

$$FFN(\mathbf{X}) = \mathbf{W}_2 \operatorname{ReLU}(\mathbf{W}_1 \mathbf{X} + \mathbf{b}_1) + \mathbf{b}_2, \quad (15)$$

where  $\mathbf{W}_1$ ,  $\mathbf{b}_1$  and  $\mathbf{W}_2$ ,  $\mathbf{b}_2$  are the parameter matrices and biases of two linear layers.

Finally, in order to enhance the integration of positioning results between the SPP algorithm and the position estimation outputs of the intelligent model, we define the output of the D-Tran model as the position increment  $\Delta P_g$  and the model confidence  $\eta$ , see as (16).

$$\left(\Delta P_{q},\eta\right) = D - \operatorname{Tran}\left(\mathbf{S}\right). \tag{16}$$

However, a significant challenge arises within supervised learning frameworks, where acquiring labels for position changes can be accomplished through additional highprecision sensor devices, while generating labels for confidence estimation proves difficult for training purposes. Consequently, we have devised a novel method for calculating the loss function. By minimizing (17) during neural network training, we achieve a better alignment with the ground truth without the necessity of confidence labels, thus resulting in enhanced feedback correction of the network.

$$\mathscr{L} = \sum_{i} \left\| \left( (1 - \eta) \Delta P_g + \eta \Delta P_S \right) - \Delta P_T \right\|^2, \quad (17)$$

where  $\Delta P_T$  is the ground truth and  $\Delta P_S$  is the output of the SPP algorithm. After estimating the input gradient on each node via backpropagation from the output, we use a gradient descent optimization method to obtain the optimal parameters that minimize the loss function. This step is crucial in improving the performance of the model by iteratively adjusting the parameters to reduce the difference between the predicted output  $\Delta P_q$  and the ground truth  $\Delta P_T$ .

During the training phase, trajectory data collected using high-precision devices are used as ground truth. The T-SPP algorithm first checks if the SPP algorithm generates an output. When the SPP algorithm provides positioning results, the trained confidence is employed to dynamically fuse the outputs of the SPP algorithm and the intelligent model, enabling the inference of the current positioning result. In scenarios where the SPP algorithm fails to produce positioning results, the current positioning result is calculated using the position change estimation derived from the output of the intelligent model. The training procedure of the D-Tran model is depicted in the pseudocode format in Algorithm 1, while Algorithm 2 depicts the real-time positioning process of the T-SPP method. The performance of our approach will be evaluated in the following section:

#### 4. Experiments

In this section, we present an analysis of the positioning performance of the T-SPP method in a typical urban canyon environment. To further investigate the adaptability of the proposed algorithm's global performance, we compare the positioning results of the traditional SPP method, DNNbased correction method, proposed T-SPP method, and factor graph method. Additionally, we analyze the performance of raw measurements from smartphone in different realistic environments.

4.1. Data Collection. This section focuses on the analysis of GNSS observation data performance in complex urban environments. To achieve this, we use the UrbanNav dataset [42] collected in typical urban canyons of Hong Kong, which provides a challenging data source due to high-rising buildings, dynamic urban canyons, and narrow streets.

Furthermore, to supplement the analysis of the UrbanNav dataset, we designed a robotic car for collecting GNSS data of smartphones and ground truth data in a medium urban environment. The framework of the robotic car is depicted in Figure 4. The smartphone was mounted on the robotic car to collect GNSS data while driving in the medium urban environment around the Institute of

Computing Technology, Chinese Academy of Sciences. On 3 March 2022, we collected GNSS data using a Xiaomi MI 8 phone with the "Geo++ RINEX Logger" app. To obtain precise location information for the smartphones, we utilized a real-time kinematic (RTK) GNSS/INS integrated solution from NovAtel SPAN-CPT, which has centimeterlevel accuracy.

4.2. Observation Quality Analysis. Analyzing the quality of raw GNSS measurements is a crucial step in isolating irrelevant features and enhancing the effectiveness of D-Tran model training. In this section, we discuss several measures, such as C/N0, pseudorange residuals, and the number of satellites observed, along with some samples. The medium urban positioning data used in this study are collected using our robocar introduced in the last section, and the UrbanNav dataset is utilized as complex data.

4.2.1. Visible Satellites. The availability of satellites in different environments is shown in Figure 5. As depicted in the figure, 8 GPS satellites were observed in urban canyon and medium urban environment using Xiaomi MI 8. However, it is important to note that the identification and total numbers of satellites vary due to the different time and place of observation data. Furthermore, the pseudorange observation data are more continuous in the medium urban environment as compared to the urban canyon. This is because deep canyon environments have more high-rising buildings, leading to numerous NLOS receptions and multipath effects, which can significantly affect the quality of GNSS measurements.

4.2.2. C/N0 of Satellites. The C/N0 is a measure of the strength of the received signal in relation to the noise level, and it is related to the different gains and losses along the entire transmitting chain. The C/N0 values obtained for the kinematic scenario while driving in different environments are presented in Figure 6. The results show that the C/N0 in the complex urban environment is at least 5 dBHz lower on average than that in the medium urban environment. This is because the quality of the received signal depends on the quality of the antenna and the reception area of the antenna, and in complex environments, smartphones with low-cost linearly polarized antennas have a smaller reception area. C/N0 is often used to evaluate the quality of satellite data, with lower C/N0 values indicating poorer satellite quality.

4.2.3. Pseudorange of Satellites. Pseudorange measurement is a crucial aspect of satellite navigation systems. However, the accuracy of the pseudorange measurements is often affected by various error sources, including receiver clock errors, multipath effects, and pseudorange noise. These errors can be quantified by the pseudorange residuals, which can be expressed mathematically using (18). To illustrate the impact of environmental factors on pseudorange residuals, Figure 7 presents the pseudorange residuals of Xiaomi 8 in both moderate and complex environments.

Input: input data S, learning rate, hyper-parameters					
Output: a well-trained D-Tran model.					
(1) initializing parameters of the D-Tran model					
(2) normalize S					
(3) calculate $\Delta P_s$					
(4) calculate $\Delta P_T$					
(5) <b>for</b> training episode $i = 1$ , N:					
(6) predict $\Delta P_a$ and $\eta$ with the inputs of S					
(7) calculate loss based on $\Delta P_q$ and $\eta$					
(8) calculate gradient					
(9) update parameters of DAE					
(10) update parameters of Transformer					
(11) end for					
(12) <b>return</b> the well-trained D-Tran model					

ALGORITHM 1: D-Tran model training.



ALGORITHM 2: Real-time positioning with T-SPP methods.



FIGURE 4: Robocar and its surveying receiver and android smartphone.

$$h = \rho_r^s - \left(\widehat{P}_r^s + I_r^s + T_r^s + c\delta t_r - c\delta t^s\right).$$
(18)

From the plot in Figure 7, it can be observed that the pseudorange residuals for both moderate and complex environments fluctuate around a zero mean, indicating that the system bias has been effectively eliminated. Nevertheless, the residuals are still influenced by observation quality, including multipath effects and clock offset errors. The results reveal that the pseudorange residuals vary significantly in moderate and complex environments, ranging from -40 to 60 meters and -80 to 60 meters, respectively. It is noteworthy that in complex environments, where tall buildings can cause severe blockage, there are some extremely large residuals outside the typical range of -80 to 60 meters. Thus, it is essential to account for these environmental factors when analyzing the accuracy of pseudorange measurements.

Table 1 presents the ranging root-mean squared error (RMSE) and maximum error of the pseudorange residuals. It can be seen from the statistics that the pseudorange accuracy



FIGURE 5: The visibility of satellites in different environments. (a) Complex environment. (b) Medium environment.



FIGURE 6: The C/N0 of the satellite signal received by the driving car in different environments. (a) Complex environment. (b) Medium environment.



FIGURE 7: The pseudorange residual in different environments. (a) Medium environment. (b) Complex environment. (c) Part of complex environment.

is approximately 46.27 m in a complex environment, whereas it is approximately 9.09 m in a moderate environment. This indicates that the antimultipath capability of smartphones is weakened, which is related to the performance of their own antenna and receiver, making them

easily influenced by the surrounding environment. Additionally, there are numerous pseudorange gross errors in smartphones, particularly in complex environments. Based on the aforementioned data analysis, it is crucial to obtain reliable positioning results in complex urban environments.

TABLE 1: Statistics of the pseudorange residuals.

Environment	Max (m)	RMSE (m)
Complex	3310.53	46.27
Medium	63.03	9.09

In order to evaluate the performance of the T-SPP algorithm in such environments, the UrbanNav dataset was utilized in this study.

4.3. Experimental Setup. The performance and continuity of the proposed T-SPP algorithm on positioning is validated by UrbanNav dataset collected in an urban area of Hong Kong. The smartphone GNSS observation data are collected using Xiao MI 8 in the RINEX format. The GNSS chip receiver of smartphone outputs signal observables at approximately 1 Hertz in our experiment methods. The ground truth is obtained from the GNSS/INS integrated solution from NovAtel SPAN-CPT. The experimental data were collected in an urban environment, with duration of 3367 seconds, about 4.86 km, where four-fifths of data are used for training, and the rest is used as the test set for performance evaluation. To evaluate the availability of our proposed T-SPP algorithm on a different smartphone, we also use the model to test with a completely new test data set collected by Huawei P40. The D-Tran model network is constructed and trained based on Python in PyTorch library, which includes the activation function (ReLU function in this study).

In the following section, we analyze and discuss the positioning accuracy and availability of the T-SPP algorithm. To address the challenge of SPP algorithms failing to provide positioning results in complex urban environments, we propose a novel evaluation metric that measures the proportion of available positioning results time over the total time.

To evaluate the effectiveness of the proposed T-SPP algorithm, we compare it with three methods on the same data sets.

- (1) T-SPP: using our proposed T-SPP algorithm based on pseudorange for the GNSS positioning task.
- (2) FGO [27]: GNSS positioning based on the integration of pseudorange and Doppler measurements using FGO.
- (3) RTKLIB: traditional SPP algorithm using pseudorange measurements.
- (4) DNN [36]: GNSS positioning by applying DNNbased correction to an initial position guess. Comparison with different networks.

Figure 8 displays a subset of the test dataset used to validate our algorithm. Traditional SPP algorithms exhibit fragility in urban environments, failing to provide positioning results when surrounded by tall buildings on both sides. On the other hand, the performance improvement of the DNN algorithm relies on high-quality satellite observation data, which is challenging to obtain in complex environments. Consequently, the DNN algorithm struggles to learn effective information, leading to limited optimization benefits. Meanwhile, the FGO algorithm can offer relatively continuous positioning results and optimize the outcome even in the presence of tall buildings on one side. However, in situations involving turns or tall buildings on both sides, insufficient optimization data reduce the availability of the FGO algorithm. Research indicates that increasing the optimization window of the FGO algorithm can enhance its performance but at the expense of increased computation time. Under the condition of using the same optimization time window, the proposed T-SPP algorithm achieves better results.

Figure 9 illustrates the positioning error curves for the four algorithms. It is evident that in open areas, all four methods demonstrate relatively small positioning errors. However, in scenarios with significant obstacles where the positioning errors of the four algorithms increase, the T-SPP algorithm exhibits superior positioning performance.

4.4. Comparison with Different Networks. We evaluate the positioning performance with different deep learning models, including the feedback neural network LSTM and the feedforward neural networks CNN. As illustrated in Figure 10, CNN and transformer can more effectively learn the characteristics of the satellite spatial structure in comparison to LSTM. Specifically, LSTM prioritizes temporal correlation in the analysis of time series, whereas it places less emphasis on spatial structure. Conversely, the transformer emphasizes the interrelationship between each row of data, thereby better utilizing the spatiotemporal correlation inherent in GNSS raw measurement.

The results of error analysis from Table 2 indicate that the transformer outperforms LSTM and CNN in terms of positioning. We conducted tests to determine the time required for predicting positioning results using the transformer model. Despite the relatively long training time required for the transformer method, it can output positioning results within 10 ms during the prediction phase. Given the growing prevalence of intelligent chips in smartphones and the incorporation of cloud-edge technologies, our algorithm can be applied to smartphone positioning during the offline prediction phase.

In our ablation experiments using UrbanNav data, the results are shown in Figure 11. The red dashed line represents the use of only the transformer model, while the blue solid line represents our proposed D-Tran model. It is evident that when using the transformer model, the fusion with the D-Tran model significantly improves accuracy compared to using the transformer model alone. We attribute this improvement to the substantial noise present in GNSS observation data. Without proper noise processing, noise can adversely affect feature extraction in the model, impeding effective feature learning from observational data during training and introducing negative impacts.

4.5. Result Analysis. Table 3 compares the performance of the T-SPP algorithm with the traditional RTKLIB method and the FGO method. The results show that the T-SPP



FIGURE 8: The experiment trajectory of different methods.



FIGURE 9: The positioning error of different methods.



FIGURE 10: The cumulative probability of position using different networks.

TABLE 2: Performance of different networks.



FIGURE 11: The cumulative probability of the D-Tran model and the Tran model.

TABLE 3: Performance of different methods.

Methods	1σ (m)	2σ (m)	RMS (m)	Positioning availability (%)
T-SPP	16.33	23.32	18.01	100
FGO	35.40	54.29	30.92	100
RTKLIB	37.12	73.22	34.60	60.84
DNN	34.71	66.84	34.18	60.84

algorithm outperforms the RTKLIB and FGO methods by 61.01% and 29.77%, respectively. Additionally, the availability of the T-SPP and FGO methods is recorded as being 100%, whereas the availability of the RTKLIB and DNN methods is only 60.84%.

As shown in Table 3 and Figure 12, the performance of the position correction method-based DNN is worse than FGO and T-SPP. This is because that this method assumes that the training dataset of observation GNSS is obtained in the open sky, without considering the impact of observation data quality on positioning in complex environments. However, in complex environments, the quality of observation data is fragile, and the method that does not consider the data quality is unlikely to achieve good optimization results in practical urban environments. The FGO method optimizes the current positioning results using historical data within a certain size window range. It can be seen that FGO can achieve 100% availability. When optimization window size is expanded, the positioning performance of



FIGURE 12: The cumulative probability of position using different methods.

FGO is often improved with the increase of the time cost. As can be seen from Figure 12, when the GNSS observation is poor for a long time, the performance of FGO is worse than T-SPP.

Considering that T-SPP and FGO have supplemented a lot of epochs, Table 4 shows the position error statistics of the four methods in the epochs that also have positioning results. It can be seen that our method still has the best performance.

We also tested the performance of the T-SPP method using a poor dataset collected from Huawei P40, and the performance of methods is shown in Table 5. We no longer compare the performance of the DNN method because we found that the performance of DNN is worse than other methods. The FGO method uses the full data to optimize the current result because we found that setting the optimizing window size to 150 cannot give reliable localization results. As can be seen in Table 5, when the quality of observation data is poor, the availability of the RTKLIB method is only 35.06% using the Huawei dataset. Although the FGO method achieved 100% reliability, it was difficult to optimize the current positioning result when the observation data quality was poor, as the optimization was based on the measurement and positioning results within the window, which could not provide good quality results when the data within the optimization window were of poor quality. Leveraging the formidable learning capability of the D-Tran network, the T-SPP algorithm exhibits superior performance when contrasted against the FGO and RTKLIB methodologies, and T-SPP can provide relatively reliable positioning results even when the quality of GNSS raw measurement is poor.

In order to further evaluate the effectiveness of the T-SPP algorithm in novel and dynamic environments, a rigorous test was conducted on the T-SPP algorithm using the GNSS

TABLE 4: Performance of different methods on the UrbanNav dataset.

Methods	1σ (m)	2σ (m)	RMS (m)
T-SPP	14.47	35.11	17.22
FGO	20.59	48.13	24.49
RTKLIB	37.12	73.22	34.60
DNN	34.71	66.84	34.18

 TABLE 5: Performance of different methods on the Huawei P40 dataset.

Methods	1σ (m)	2σ (m)	RMS (m)	Positioning availability (%)
T-SPP	33.65	110.23	54.77	100
FGO	45.26	131.92	93.64	100
RTKLIB	49.61	472.61	127.48	35.06

TABLE 6: Performance of different methods on the Beijing dataset.

Methods	1σ (m)	2σ (m)	RMS (m)	Positioning availability (%)
T-SPP	6.22	10.67	6.77	100
FGO	7.09	12.38	9.02	100
RTKLIB	13.38	27.43	13.73	83.89

observation dataset collected by Didi Corporation in Beijing. The positioning errors of our proposed method are shown in Table 6. As can be seen from Table 6, the T-SPP also outperforms other comparative baselines. T-SPP and FGO maintain an error within ten meters in the urban environment of Beijing and can position the current road without crossing the road network structure. A better effect can be achieved by combining the road network topology structure. Compared to FGO, T-SPP achieves better positioning accuracy. While the transformer model utilized in our study demands a substantial training time investment, it exhibits an efficient computational performance during testing, requiring only 20 ms for the computation of each output.

#### 5. Conclusion

This study proposes a novel sequence-to-sequence positioning algorithm that combines the GNSS SPP algorithm with the D-Tran model. Firstly, the GNSS raw observation data are denoised using DAE and then the transformer model is used to capture the spatial correlations in the GNSS position sequence. This method automatically learns latent features from the GNSS observation sequence, filling occasional or short-term missing GNSS position data. The model simultaneously outputs position increments and model confidence estimates, adaptively adjusting the fusion weights of the model and SPP algorithm to optimize the positioning results. Experimental results on multiple datasets demonstrate that the T-SPP method can provide continuous positioning results in challenging environments and outperforms FGO and traditional methods.

Although the T-SPP method shows promising prospects in SPP positioning, it cannot replace classical geometricbased methods. On the contrary, combining geometricbased methods with representations, knowledge, and models learned through transformers will serve as a feasible complement to further improve the accuracy and availability of SPP systems. The inherent opaqueness of deep learning models, rendering them as black-box systems with challenging interpretability of internal workings, underscores the caution required when considering their substitution for classical geometry-based approaches. This paper advocates the integration of geometric methodologies with representations, knowledge, and models acquired through transformer learning, presenting a viable supplementation to enhance the accuracy and availability of SPP systems. The proposed T-SPP method, grounded in deep learning, demands substantial computational resources during the training phase, posing challenges in environments constrained by device capabilities. It is conceivable that future advancements in hardware technology may pave the way for the feasibility of deep learning methods on mobile devices [43].

## **Data Availability**

The raw data supporting the conclusions of this article will be made available by the authors.

### **Conflicts of Interest**

The authors declare that they have no conflicts of interest.

# Acknowledgments

This work was supported in part by the National Key Research and Development Program under grant no. 2022YFB3904700, the National Natural Science Foundation of China under grant numbers 62261042 and 62002026, the Key Research Projects of the Joint Research Fund for Beijing Natural Science Foundation and the Fengtai Rail Transit Frontier Research Joint Fund under grant no. L221003, Beijing Natural Science Foundation under grant numbers 4212024 and 4222034, the Strategic Priority Research Program of Chinese Academy of Sciences under grant no. XDA28040500, the Fundamental Research Funds for the Central Universities under grant no. 2022RC13, and the BUPT Excellent Ph.D. Students Foundation under grant no. CX2022131.

# References

- F. Dovis, L. Ruotsalainen, R. Toledo-Moreo, Z. Z. M. Kassas, and V. Gikas, "Recent advancement on the use of global navigation satellite system-based positioning for intelligent transport systems [guest editorial]," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 6–9, 2020.
- [2] A. Gholamhosseinian and J. Seitz, "Vehicle classification in intelligent transport systems: an overview, methods and software perspective," *IEEE Open Journal of Intelligent Transportation Systems*, vol. 2, pp. 173–194, 2021.

- [3] N. M. Alsubaie, A. A. Youssef, and N. El-Sheimy, "Improving the accuracy of direct geo-referencing of smartphone-based mobile mapping systems using relative orientation and scene geometric constraints," *Sensors*, vol. 17, no. 10, p. 2237, 2017.
- [4] L. T. Hsu, "GNSS multipath detection using a machine learning approach," in *Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems* (*ITSC*), pp. 1–6, IEEE, Yokohama, Japan, October 2017.
- [5] Z. Gong, R. Ying, F. Wen, J. Qian, and P. Liu, "Tightly coupled integration of GNSS and vision SLAM using 10-DoF optimization on manifold," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12105–12117, 2019.
- [6] R. Sun, Y. Yang, K. W. Chiang, T. T. Duong, K. Y. Lin, and G. J. Tsai, "Robust IMU/GPS/VO integration for vehicle navigation in GNSS degraded urban areas," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10110–10122, 2020.
- [7] T. Shan and B. Englot, "Lego-loam: lightweight and groundoptimized lidar odometry and mapping on variable terrain," in *Proceedings of the 2018 IEEE/RSJ International Conference* on *Intelligent Robots and Systems (IROS)*, pp. 4758–4765, IEEE, Madrid, Spain, October 2018.
- [8] Q. Chen, Q. Zhang, and X. Niu, "Estimate the pitch and heading mounting angles of the IMU for land vehicular GNSS/INS integrated system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6503– 6515, 2021.
- [9] A. S. El-Wakeel, A. Osman, N. Zorba, H. S. Hassanein, and A. Noureldin, "Robust positioning for road information services in challenging environments," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3182–3195, 2020.
- [10] E. G. S. A. World's, *First Dual-Frequency GNSS Smartphone Hits the Market*, European Global navigation Satellite System Agency, Prague, Czech Republic, 2018.
- [11] J. Geng, E. Jiang, G. Li, S. Xin, and N. Wei, "An improved Hatch filter algorithm towards sub-meter positioning using only Android raw GNSS measurements without external augmentation corrections," *Remote Sensing*, vol. 11, no. 14, p. 1679, 2019.
- [12] M. Harris, P. Schlais, T. Murphy, A. Joseph, and J. Kazmierczak, "GPS and GALILEO airframe multipath error bounding method and test results," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 114–139, Miami, FL, USA, September 2020.
- [13] J. Blanch, E. Phelts, K. Gunning, T. Walter, L. de Groot, and L. Norman, "Bounding temporally correlated measurement noise with an application to GNSS integrity," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 27–38, Miami, FL, USA, September 2020.
- [14] W. Wen, Y. Zhou, G. Zhang et al., "Urbanloco: a full sensor suite dataset for mapping and localization in urban scenes," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2310–2316, IEEE, Paris, France, May 2020.
- [15] G. Zhang, P. Xu, H. Xu, and L. T. Hsu, "Prediction on the urban GNSS measurement uncertainty based on deep learning networks with long short-term memory," *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20563–20577, 2021.
- [16] F. Wu, H. Luo, H. Jia, F. Zhao, Y. Xiao, and X. Gao, "Predicting the noise covariance with a multitask learning model for Kalman filter-based GNSS/INS integrated navigation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.

- [17] W. Fang, J. Jiang, S. Lu et al., "A LSTM algorithm estimating pseudo measurements for aiding INS during GNSS signal outages," *Remote Sensing*, vol. 12, no. 2, p. 256, 2020.
- [18] X. Gao, H. Luo, B. Ning et al., "RL-AKF: an adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles," *Remote Sensing*, vol. 12, no. 11, p. 1704, 2020.
- [19] Y. Xiao, H. Luo, F. Zhao et al., "Residual attention networkbased confidence estimation algorithm for non-holonomic constraint in GNSS/INS integrated navigation system," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 11404–11418, 2021.
- [20] N. Zhu, D. Betaille, J. Marais, and M. Berbineau, "GNSS integrity monitoring schemes for terrestrial applications in harsh signal environments," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 81–91, 2020.
- [21] W. Shuai, Q. Xing, X. Li, Y. Xing, and X. Cai, "An improved marine multi-constellation pseudorange single-point positioning algorithm," in *Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Paris)*, pp. 1–6, IEEE, Paris, France, May 2019.
- [22] D. Prochniewicz, K. Wezka, and J. Kozuchowska, "Empirical stochastic model of multi-GNSS measurements," *Sensors*, vol. 21, no. 13, p. 4566, 2021.
- [23] S. Z. Farooq, X. Hong, D. Yang, M. Xia, and E. N. J. Ada, "Point positioning using extended Kalman filtering in single frequency Beidou receivers," in *Proceedings of the 2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, pp. 1–5, IEEE, Wuhan, China, March 2018.
- [24] L. Guo, F. Wang, J. Sang, X. Lin, X. Gong, and W. Zhang, "Characteristics analysis of raw multi-GNSS measurement from Xiaomi Mi 8 and positioning performance improvement with L5/E5 frequency in an urban environment," *Remote Sensing*, vol. 12, no. 4, p. 744, 2020.
- [25] B. Chen, C. Gao, Y. Liu, and P. Sun, "Real-time precise point positioning with a Xiaomi MI 8 android smartphone," *Sensors*, vol. 19, no. 12, p. 2835, 2019.
- [26] X. Yang and N. Wang, "Precise single point positioning algorithm of GPS and BDS fusion," in *Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, IEEE, Chongqing, China, December 2020.
- [27] W. Wen and L. T. Hsu, "Towards robust GNSS positioning and Real-time kinematic using factor graph optimization," in *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5884–5890, IEEE, Xi'an, China, June 2021.
- [28] N. Linty, A. Farasin, A. Favenza, and F. Dovis, "Detection of GNSS ionospheric scintillations based on machine learning decision tree," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 303–317, 2019.
- [29] M. Kaselimi, A. Voulodimos, N. Doulamis, A. Doulamis, and D. Delikaraoglou, "Deep recurrent neural networks for ionospheric variations estimation using gnss measurements," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.
- [30] A. Elango, S. Ujan, and L. Ruotsalainen, "Disruptive GNSS signal detection and classification at different power levels using advanced deep-learning approach," in *Proceedings of the* 2022 International Conference on Localization and GNSS (ICL-GNSS), pp. 1–7, Tampere, Finland, June 2022.
- [31] I. E. Mehr and F. Dovis, "Detection and classification of GNSS jammers using convolutional neural networks," *IEEE*, in

Proceedings of the 2022 International Conference on Localization and GNSS (ICL-GNSS), pp. 01–06, Tampere, Finland, June 2022.

- [32] E. Munin, A. Blais, and N. Couellan, "Convolutional neural network for multipath detection in GNSS receivers," *IEEE*, in *Proceedings of the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–10, Singapore, September 2020.
- [33] S. J. Cho, B. S. Kim, T. S. Kim, and S. H. Kong, "Enhancing GNSS performance and detection of road crossing in urban area using deep learning," in *Proceedings of the 2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 2115–2120, Auckland, New Zealand, October 2019.
- [34] E. Munin, A. Blais, and N. Couellan, "Convolutional neural network for multipath detection in GNSS receivers," in *Proceedings of the 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*, pp. 1–10, Auckland, New Zealand, February 2020.
- [35] D. Min, M. Kim, J. Lee, and J. Lee, "Deep neural network based multipath mitigation method for carrier based differential GNSS systems," in *Proceedings of the ION 2019 Pacific PNT Meeting*, pp. 451–466, Honolulu, Hawaii, April 2019.
- [36] A. V. Kanhere, S. Gupta, A. Shetty, and G. Gao, "Improving GNSS positioning using neural network-based corrections," 2021, https://arxiv.org/abs/2110.09581.
- [37] F. Guo, X. Zhang, and J. Wang, "Timing group delay and differential code bias corrections for BeiDou positioning," *Journal of Geodesy*, vol. 89, no. 5, pp. 427–445, 2015.
- [38] J. A. Klobuchar, "Ionospheric time-delay algorithm for singlefrequency GPS users," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 3, pp. 325–331, 1987.
- [39] J. Saastamoinen, "Contributions to the theory of atmospheric refraction," *Bulletin Geodesique*, vol. 105, no. 1, pp. 279–298, 1972.
- [40] J. M. Dow, R. E. Neilan, and C. Rizos, "The international GNSS service in a changing landscape of global navigation satellite systems," *Journal of Geodesy*, vol. 83, no. 7, p. 689, 2009.
- [41] Y. T. Chan and J. J. Towers, "Sequential localization of a radiating source by Doppler-shifted frequency measurements," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 28, no. 4, pp. 1084–1090, 1992.
- [42] A. Dosovitskiy, L. Beyer, A. Kolesnikov et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, https://arxiv.org/abs/2010.11929.
- [43] L. T. Hsu, N. Kubo, W. Wen et al., "UrbanNav: an opensourced multisensory dataset for benchmarking positioning algorithms designed for urban areas," in *Proceedings of the* 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), pp. 226–256, St. Louis, Missouri, September 2021.