Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deep Reinforcement Learning with Robust Spatial-Temporal Representation for Improving GNSS Positioning Correction

Zhenni Li
Guangdong University of Technology
Peili Li
3122000799@mail2.gdut.edu.cn
Guangdong University of Technology
Jianhao Tang
Guangdong University of Technology
Yulong Song
Guangzhou Haige Communications Group Incorporated Company
Liji Chen
Guangdong University of Technology
Yiting Cai
Guangdong University of Technology
Shengli Xie
Ministry of Education

Research Article

Keywords: GNSS positioning, deep reinforcement learning, self-supervised learning, generalization, transformer

Posted Date: August 5th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-7221530/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Deep Reinforcement Learning with Robust Spatial-Temporal Representation for Improving GNSS Positioning Correction

Zhenni Li[1,2], Peili Li[1,3*], Jianhao Tang[1,3], Yulong Song[4], Liji Chen[1,3], Yiting Cai[1,3], Shengli Xie[3,5]

[1] School of Automation, Guangdong University of Technology, Guangzhou, 510006, China.

[2] Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing (GDUT), Guangzhou 510006, China.

[3] Guangdong Provincial Key Laboratory of Intelligent Systems and Optimization Integration, Guangzhou 510006, China.

[4] Guangzhou Haige Communications Group Incorporated Company, Guangzhou 510006, China.

[5] Key Laboratory of Intelligent Information Processing and System Integration of IoT（GDUT）, Ministry of Education, Guangzhou 510006, China.

*Corresponding author. E-mail: 3122000799@mail2.gdut.edu.cn.

Abstract

In complex urban environments, where GNSS positioning is severely degraded by multipath interference and non-line-of-sight reception, data-driven methods offer a promising solution by effectively modeling complex non-Gaussian errors from sufficient data for positioning correction. The inherent spatial geometric relationships among different constellations in single-epoch GNSS observations, and the temporal dependencies exhibited in sequential multi-epoch observations, contain rich spatial-temporal information that facilitates the modeling of complex stochastic noise in GNSS measurements. However, the effective extraction and correlation of these multidimensional features from GNSS observation data have not yet been sufficiently explored in existing studies. Moreover, dynamic changes in real-world environments induce data distribution shift between training and testing, requiring generalization capability for the data-driven model in unseen scenarios. In this paper, we propose a

novel deep reinforcement learning model with robust spatial-temporal representation (DRL-RSTR) for GNSS positioning correction. The spatial geometric relationships among different constellations is modeled by a graph convolutional network (GCN), and the temporal dependencies of sequential observations are captured by transformer. Then, the spatial-temporal features are fused through summation, and a cross-attention network is employed to model the interactions among multi-observations to obtain a comprehensive environmental representation. Finally, we construct a multi-observation GCN-transformer (MOGT) to encode spatial-temporal representation. Additionally, a self-supervised pretext task (SST) is introduced to improve the robustness of spatial-temporal representation against data distribution shift through consistency regularization across non-augmented and augmented observations. We conduct extensive experiments on the public GSDC and built GZGNSS datasets, results show that DRL-RSTR achieves superior positioning accuracy and generalization compared to the model-based and learning-based state-of-the-art methods, with improvements of 51.2% and 41.4% on the GZGNSS dataset and 6.5% compared with kalman filters on the GSDC dataset in terms of positioning accuracy.

Keywords GNSS positioning, deep reinforcement learning, self-supervised learning, generalization, transformer

Introduction

With the extensive use of the Global Navigation Satellite System (GNSS) in autonomous driving applications, the demand for high-accuracy position-based services has increased dramatically (Ahmad et al., 2020; Yue et al.,2024). While GNSS positioning performance is remarkable in open areas (Niu et al., 2025), it still faces challenges in different complex urban scenarios, such as overpasses, urban canyons, urban forests, etc. As shown in Fig. 1, due to the effects of multipath interference and non-line-of-sight (NLOS) reception (Breßler et al., 2016), the positioning error can be tens of meters (Hsu, 2018). To address this problem, numerous methods have been developed to improve the accuracy of GNSS positioning. Common model-based methods include the weighted least squares (WLS) Medina et al., 2019), Kalman filters

(KF) (Kalman, 1960), real-time kinematic (RTK) (Wang, 1999; Eissfeller et al., 2001; Li et al., 2022), etc. For instance, the KF improves positioning accuracy in noisy urban environments by optimally estimating the system state through properly tuning the noise covariance matrix. However, their effectiveness is highly dependent on the prior assumptions on environmental parameters and model hyperparameters. This dependency can limit their practicality in complex non-stationary environments.

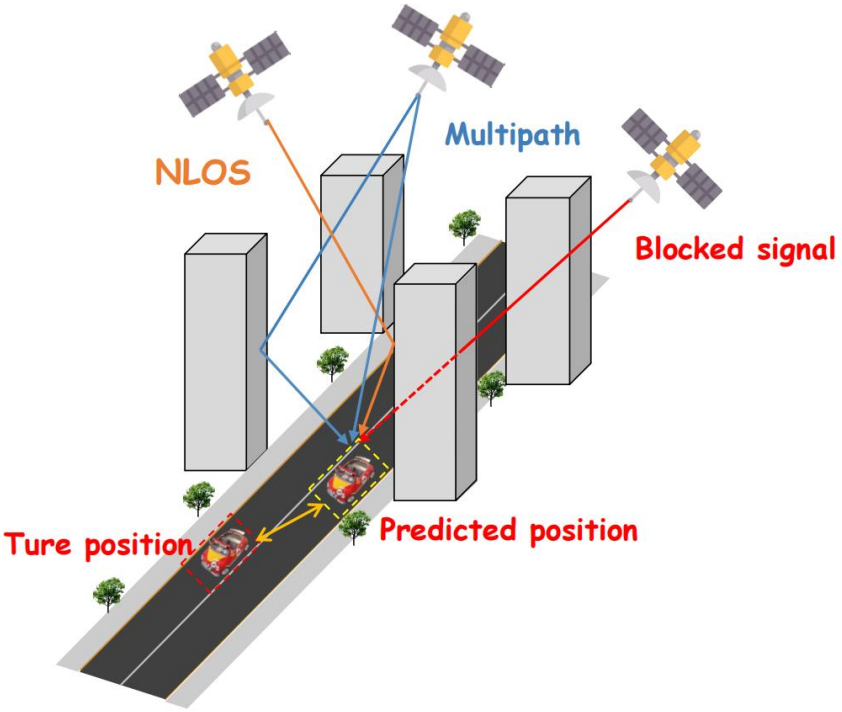


Fig. 1 The vehicle positioning has been severely offset due to the effects of multipath interference and NLOS reception.

Learning-based methods have emerged as novel GNSS positioning augmentation model (Mohanty and Gao, 2024). These approaches can model complex urban environmental errors by learning from a large amount of data. In recent years, many deep learning (DL)-based studies have been proven to improve the accuracy of GNSS positioning. These methods utilize neural networks to predict position corrections for correcting initial positions (Kanhere et al., 2022; Mohanty and Gao, 2023; Weng et al., 2024; Liu et al., 2022, 2019). However, the surrounding environment of moving vehicles can change rapidly in non-stationary urban areas. DL-based methods only consider the learning for each position discretely, ignoring the sequential relationship between positions. Deep reinforcement learning (DRL)-based methods leverage the mechanism of agent-environment interaction to learn a position-corrected policy based

on the cumulative rewards along the positioning trajectory, helping the model consider the sequential relationship between positions. For example, in Tang et al.,2024, the researchers employ a long short-term memory (LSTM) module with reinforcement learning (RL) to extract temporal features from GNSS observations. In Zhao et al., 2023b, the authors consider the relationship between different constellation satellites and utilizes a graph neural network (GNN) with RL to extract spatial features.

Existing DRL-based studies have achieved remarkable performance in improving GNSS positioning (Zhao et al., 2023a; Li et al., 2024; Tang et al., 2025; Han et al., 2021; Shin et al., 2018; Gao et al., 2020). Despite their effectiveness, current methods still have some problems: 1) single-epoch GNSS observations naturally encapsulate spatial geometric relationships among different constellation satellites. Furthermore, satellite movement induces time-varying changes in signal incidence status, leading to dynamic changes in positioning errors. These features can be inherently captured through sequential multi-epoch observations. However, existing methods mainly focus on isolated spatial or temporal feature extraction, ignoring the spatial-temporal correlation, which is crucial for the model to comprehend the complex environment and improve its position-corrected abilities. Extracting and correlating the rich spatial-temporal features from GNSS observations remains unexplored; 2) The success of these methods depends on a significant assumption that the training and testing data are independently distributed. In practice, the dynamic changes in the surrounding environment of the moving vehicle can cause data distribution shifts, which cause performance degradation in unseen urban scenarios.

To address these problems, we propose a novel DRL-based model with robust spatial-temporal representation (DRL-RSTR) to improve GNSS positioning correction and generalize to unseen environments. To extract the spatial-temporal correlations from GNSS multi-observation, we propose a multi-observation GCN-transformer (MOGT). MOGT includes a GCN module for extracting spatial features, a transformer module for extracting temporal features, spatial-temporal features fusion through summation, and a cross-attention network for multi-observation information fusion. Moreover, to tackle the problem of data distribution shift in dynamic changing environments, we propose a self-supervised pretext task (SST) to improve the

robustness of spatial-temporal representation. It first generates augmented data through data augmentation to model dynamic changes in the driving scene. Then, it aims at learning invariant representation that remains robust to dynamic changes by enforcing consistency regularization across non-augmented and augmented observations, thereby improving model generalization. With the MOGT and the SST, we develop a novel DRL-based model for improving GNSS positioning correction, aimed at achieving high-precision GNSS positioning correction in complex urban areas. Specially, the main contributions of our work can be outlined as follows:

1) To encode spatial-temporal representation for enhancing the model's understanding of complex environments, the MOGT is proposed to extract spatial-temporal features from GNSS observations in complex urban areas. It utilizes a GCN module for extracting spatial features and a transformer module for extracting temporal features, respectively. After a simple summation for spatial-temporal features fusion, a cross-attention network is used for modeling the interactions among multi-observation.

2) To address the performance degradation in dynamic changing environments caused by data distribution shift, we propose SST to improve the robustness of the latent representation against dynamic changes. Specifically, SST utilizes data augmentation to model dynamic changes in the driving scene and generate augmented data. Then it enforces consistency regularization to align latent representation across non-augmented and augmented observations, thereby learning invariant representation that remains robust to dynamic changes.

3) The proposed method is proven to boost performance by validating it in the public GSDC dataset and the built GZGNSS dataset. The results show that our method achieves better generalization performance to unseen environments compared to existing model-based and learning-based SOTA positioning methods, with improvements of 51.2% and 41.4% on the GZGNSS dataset and 6.5% on the GSDC dataset in terms of positioning accuracy.

Related Work

In this section, we will introduce the related work in two aspects: GNSS positioning and generalization in deep reinforcement learning for GNSS positioning correction.

GNSS Positioning

The GNSS positioning methods can be broadly classified into two categories: model-based and learning-based methods. We provide a detailed introduction to these two types of methods as follows:

Model-based methods for GNSS positioning: Conventional model-based methods rely on established physical and mathematical models to process GNSS signals and determine the positions of receivers. For example, single-point positioning (SPP) methods utilize single-frequency pseudorange observation between a satellite and a receiver to derive position information, such as WLS and KF, which are typically employed for smartphone positioning. Precise point positioning (PPP) can achieve static centimeter-level and dynamic meter-level positioning accuracy by using pseudorange and carrier-phase observations (Zumberge et al., 1997). RTK utilizes a reference station with a known position and a rover receiver. RTK can achieve centimeter-level positioning accuracy by using correction data from a reference station in real-time. However, model-based methods require strict assumptions on model parameters and are sensitive to multipath effects, signal obstruction, and other environmental factors in urban areas.

Learning-based methods for GNSS positioning: With the rapid development of artificial intelligence, learning-based methods provide novel approaches for improving GNSS positioning in complex environments. Compared to model-based methods, learning-based methods require fewer assumptions. Recently, more scholars have proposed using DL-based methods to model complex GNSS measurement errors for improving the accuracy of GNSS positioning. For instance, Kanhere et al. utilizes deep neural networks (DNNs) to estimate the position corrections (Kanhere et al., 2022). The DNNs learn a mapping from GNSS measurements to these position corrections, thereby improving positioning accuracy. Weng et al. proposes a novel neural network, PrNet, to mitigate pseudoranges to improve positioning accuracy (Weng et al., 2024) . However, DL-based methods did not consider the positioning continuity in complex environments. DRL demonstrates tremendous potential for improving GNSS positioning in dynamic changing environments. Zhang et al. first proposed to use RL to increase GPS positioning accuracy (Zhang and Masoud, 2020). Li et al. utilizes an

attention-LSTM module to capture historical information from the observation sequence (Li et al., 2024). Further, Zhao et al. combines GNN with RL to fuse spatial features from the different constellations (Zhao et al., 2023b). Nonetheless, these methods primarily focus on either spatial or temporal aspects of GNSS observation, ignoring the spatial-temporal correlations. In our work, we propose a multi-observation GCN-transformer framework. It learns the spatial-temporal representation of GNSS multi-observations to enhance GNSS positioning correction.

Generalization in Deep Reinforcement Learning for GNSS Positioning Correction

Since DRL-based methods have achieved remarkable breakthroughs, more and more scholars have begun to study the application of DRL-based methods in GNSS positioning correction (Zhao et al., 2024; Li et al., 2024). However, their success depends on a key assumption that the training and testing data are independently distributed. A recent study has shown that the dynamic changes in the surrounding environment of the moving vehicle potentially lead to a data distribution shift between training and testing phases (Tang et al., 2025). This problem significantly limits the generalization of trained models to unseen environments. Currently, there are fewer related studies in the field of improving generalization for GNSS position-corrected models. To address the problem of data distribution shift in dynamically changing environments, Tang et al. proposes the RBO-based PPO for policy learning (Tang et al., 2025). Li et al. propose to utilize data augmentation to increase training data and self-supervised learning to learn single-step invariant representation that is robust to dynamic changes (Li et al., 2024). In comparison, inspired by (Hansen and Wang, 2021), we design a simple self-supervised task to train the encoder to encode the robust representation for improving model generalization in dynamically changing environments. Unlike in Li et al., 2024, which focuses on learning single-step invariant representation from the current observation, our approach leverages sequential observation to learn sequential invariant representation, which is better adaptable to dynamically changing environments.

Proposed method

In this section, we first introduce the model-based method KF for initial positions. Then,

we introduce our proposed method, DRL-RSTR, to correct the initial positions, including the RL environment of the GNSS positioning correction task, MOGT for extracting spatial-temporal features, SST for encoding robust representation, and an actor-critic network using PPO for policy learning. The schematic diagram of our proposed method is shown in Fig. 2.

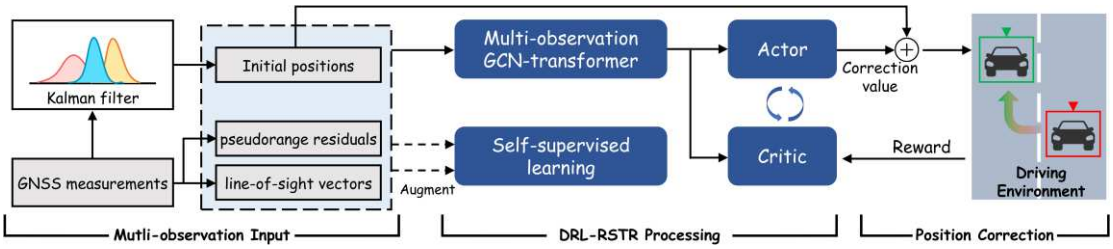


Fig. 2 The schematic diagram of our proposed method.

KF for Initial Positions

For better initialization for our proposed DRL-RSTR, we use KF as our baseline to estimate initial positions. KF is a recursive state estimation method proposed by Rudolf E (Kalman, 1960). The system model of KF can be defined as:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t, \tag{1}$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t, \tag{2}$$

where equation (1) is a state transition model and (2) is an observation model, the the state vector $\mathbf{x}_t := \mathbf{p}_t = (x_t, y_t, z_t)$ is defined as the vehicle position at timestep t, $\mathbf{u}_t$ is the control input, the actual measurement $\mathbf{z}_t := \mathbf{p}_t^{WLS} = (x_t^{WLS}, y_t^{WLS}, z_t^{WLS})$ is defined as the measured position by WLS, $\mathbf{A} \in \mathbb{R}^{3\times3}$ represents the state transition matrix, $\mathbf{B} \in \mathbb{R}^{3\times3}$ represents the control input matrix, $\mathbf{H} \in \mathbb{R}^{3\times3}$ is the observation matrix, all of the matrices are defined as identity matrices. $\mathbf{w}_t$ and $\mathbf{v}_t$ represent the process and observation noises, respectively, which follow a Gaussian distribution.

KF consists of two steps: prediction and update. Firstly, KF predicts the current state and covariance matrix from a system model with the previous state and covariance matrix.

$$\hat{\mathbf{x}}_{t|t-1} = \mathbf{A}\hat{\mathbf{x}}_{t-1|t-1} + \mathbf{B}\mathbf{u}_t, \tag{3}$$

$$\mathbf{P}_{t|t-1} = \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^{\mathrm{T}} + \mathbf{Q}, \tag{4}$$

where $\hat{\mathbf{x}}_{t|t-1}$ and $\mathbf{P}_{t|t-1}$ represent the predicted current state and error covariance matrix at timestep t, and $\mathbf{Q}$ is the covariance matrix of process noise. Then the state is

updated by the equation:

$$\mathbf{K}_t = \mathbf{P}_{t-1|t-1}\mathbf{H}^\top\left(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^\top + \mathbf{R}\right)^{-1}, \tag{5}$$

$$\hat{\mathbf{x}}_{t|t} = \hat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t\left(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}_{t|t-1}\right), \tag{6}$$

$$\mathbf{P}_{t|t} = (I - \mathbf{K}_t\mathbf{H})\mathbf{P}_{t|t-1}, \tag{7}$$

where $\mathbf{K}_t$ represents the Kalman gain obtained by the predicted error covariance matrix $\mathbf{P}_{t|t-1}$, observation matrix $\mathbf{H}$, and the observation noise covariance $\mathbf{R}$. $\hat{\mathbf{x}}_{t|t}$ represents the updated state calculated using $\hat{\mathbf{x}}_{t|t-1}$, the Kalman gain $\mathbf{K}_t$, and $\mathbf{z}_t$. $\mathbf{P}_{t|t}$ represents the updated covariance matrix calculated using $\mathbf{K}_t$ and the predicted covariance matrix. In dynamic changing environments, it is hard for KF to model the nonlinear and non-Gaussian errors caused by multipath interference and NLOS reception, and also requires strict parameter assumptions. Therefore, we propose a DRL-based method to further correct the initial positions obtained by KF.

DRL-RSTR for Positioning Correction

In this section, we introduce our DRL-based positioning correction method in four parts, including a sequential multi-observation setting, a multi-observation GCN-transformer for spatial-temporal representation, self-supervised learning for encoding robust representation, a continuous position-corrected action space setting, and a reward function setting. The framework of our proposed DRL-RSTR is shown in Fig. 3.

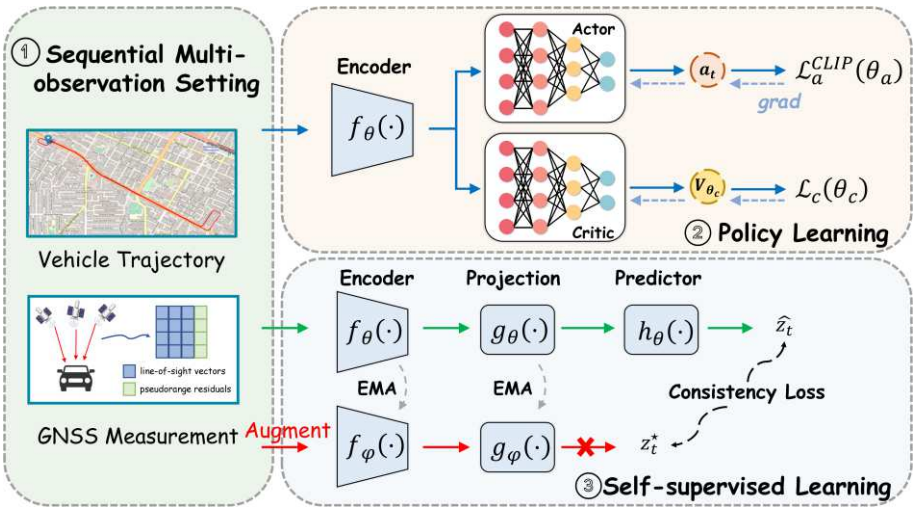


Fig. 3 The framework of the proposed DRL-RSTR consists of three key components: a sequential multi-observation setting, policy learning, and self-supervised learning. The "Encoder" module, depicted in the figure, corresponds to our MOGT model, which is detailed in Fig. 4.

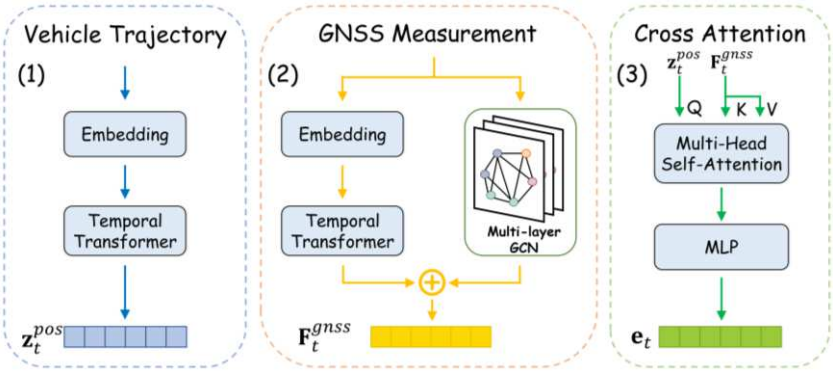


Fig. 4 The framework of our MOGT is composed of three main components:(1) Temporal feature extraction from vehicle trajectory, (2) Spatial-temporal feature extraction from GNSS measurements, and (3) Cross-attention network to model the interactions between vehicle trajectory and GNSS measurements.

Sequential multi-observation setting: Since the real-world environment is partially observable, the GNSS positioning correction task can be modeled as a Partially Observable Markov Decision Process (POMDP) $\mathcal{PM} = \langle \mathcal{O}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$. Here, $\mathcal{O}$ is the observation space, $\mathcal{A}$ is the action space, $\mathcal{T}: \mathcal{O} \times \mathcal{A} \times \mathcal{O} \to [0,1]$ is the transition dynamic of observations, $\mathcal{R}: \mathcal{O} \times \mathcal{A} \to \mathbb{R}$ is the reward function and $\gamma \to [0,1]$ is a discount factor. The RL trajectory of length T can be represented as a sequence of experience tuples $\{(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1})\}_{t=1}^{T}$. Specifically, at each timestep *t*, the vehicle agent receives current observation $\mathbf{o}_t \in \mathcal{O}$ from the environment and executes an action $\mathbf{a}_t \in \mathcal{A}$ according to its policy $\mathbf{a}_t \sim \pi(\cdot\,|\mathbf{o}_t)$. The environment then returns a reward $r_t(\mathbf{o}_t, \mathbf{a}_t)$ determined by $\mathcal{R}$, and transitions to the next observation $\mathbf{o}_{t+1}$ according to $\mathcal{T}(\mathbf{o}_{t+1}|\mathbf{o}_t, \mathbf{a}_t)$. The objective of the agent is to learn an optimal policy $\pi^\star$ to obtain the maximum expected discounted cumulative return $\mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r_t]$.

To comprehensively describe the current state of the vehicle agent, we construct a sequential multi-observation input. The sequential multi-observation input includes the vehicle trajectory $\mathcal{P}_t$ and sequential GNSS measurements $\mathcal{M}_t$ related to the multi-constellation satellites at time *t*, denoted as:

$$\mathbf{o}_t = [\mathcal{P}_t; \mathcal{M}_t], \ \forall t \in \{1, \ldots, T\}, \tag{8}$$

where $\mathcal{P}_t$ is a sequence of vehicle positions over historical horizon T_h, denoted as $\mathcal{P}_t = \{\mathbf{p}_i\}_{i=t-T_h}^{t}$ and $\mathbf{p}_i = (x_i, y_i, z_i)$ represents the three-dimensional position coordinates of the vehicle in the Earth-Centered Earth-Fixed (ECEF) coordinate system. $\mathcal{M}_t = \{\mathbf{M}_i\}_{i=t-T_h}^{t}$ is a sequence of GNSS measurements over the same historical

horizon T_h. $\mathbf{M}_i = \left(\mathbf{m}_i^1, \ldots, \mathbf{m}_i^{N_{max}}\right) \in \mathbb{R}^{N_{max} \times 4}$, where $\mathbf{m}_i^j \in \mathbb{R}^4$ is composed of line-of-sight (LOS) vector $\mathcal{LOS}_i^j \in \mathbb{R}^3$ and pseudorange residual $\mathcal{PR}_i^j \in \mathbb{R}^1$ related to the multi-constellation satellites, which can be computed as follows,

$$\mathcal{LOS}_i^j = \frac{\mathbf{sp}_i^j - \mathbf{p}_i}{\left\|\mathbf{sp}_i^j - \mathbf{p}_i\right\|_2}, \mathcal{PR}_i^j = \rho_i^j - \left\|\mathbf{sp}_i^j - \mathbf{p}_i\right\|_2$$

$$\forall t \in \{t - T_h, \ldots, t\}, j \in \{1, \ldots, N_{max}\}, \tag{9}$$

where $\mathbf{sp}_i^j$ is the estimated *j*-th satellite position at time *i* and ρ_i^j is the corresponding pseudorange. Notably, since the number of visible satellites at different times or places is constantly changing, we set a fixed upper limit for the number of visible satellites N_{max} to standardize the input observation dimension, making it easier for the model to process. The values of $\mathcal{LOS}_i^j$ and $\mathcal{PR}_i^j$ corresponding to non-visible satellites are filled with zero vectors.

Multi-observation GCN-transformer for encoding spatial-temporal representation: MOGT aims to capture the spatial-temporal correlations given sequential multi-observation. The proposed MOGT framework operates through three key stages: 1) transformer for extracting temporal features: a transformer-based architecture extracts temporal features separately from vehicle trajectory and sequential GNSS measurements. 2) GCN for extracting spatial features: a GCN network is used to extract spatial features from GNSS measurements. 3) Spatial-temporal features fusion and multi-observation information fusion: a cross-attention network is used to model the interactions among vehicle trajectory and sequential GNSS measurements. The overall framework of MOGT is shown in Fig. 4.

1) Transformer for extracting temporal features: Due to the interference of dynamic changing environments for GNSS positioning, it is vital to extract temporal features to improve GNSS positioning. Transformer has demonstrated remarkable performance across various tasks in the field of time series modeling, particularly in capturing long-range dependencies and global temporal features (Ahmed et al., 2023; Wen et al., 2022; Wu et al., 2020; Zhou et al., 2021; Nie et al., 2022; Chen et al., 2023).. Therefore, we employ transformer to encode the temporal representation of GNSS multi-observation. The Multi-head Self-Attention (MSA) (Vaswani et al., 2017) is the core mechanism in the transformer architecture, which can effectively capture long-range dependencies

between the time series data. Here we briefly introduce the MSA block as follows.

Given the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ matrices, which are obtained by linearly projecting the input vectors using learnable weight matrices $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$, the output of a single attention head is computed as:

$$\mathrm{Att}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d_k}}\right), \tag{10}$$

where d_k is the dimension of key vectors. The outputs of multiple heads h are concatenated to capture different aspects of information and passed through a linear projection module to produce the output of MSA:

$$\mathrm{MSA}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{Concat}(\mathrm{head}_1,\ldots,\mathrm{head}_h)\mathbf{W}^O, \tag{11}$$

where

$$\mathrm{head}_i = \mathrm{Att}\left(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V\right), \tag{12}$$

where the projections are parameter matrices $\mathbf{W}_i^Q \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^K \in \mathbb{R}^{d_m \times d_k}$, $\mathbf{W}_i^V \in \mathbb{R}^{d_m \times d_v}$, and $\mathbf{W}^O \in \mathbb{R}^{d_{hv} \times d_m}$. The MSA block also includes a residual connection followed by layer normalization (LN) and a feed-forward network (FFN) (Vaswani et al., 2017). Finally, the temporal transformer of a single layer can be represented as:

$$\mathrm{Trans}_t(x) = \mathrm{LN}\left(\mathrm{LN}\left(x + \mathrm{MSA}(x,x,x)\right) + \mathrm{FFN}\left(\mathrm{LN}\left(x + \mathrm{MSA}(x,x,x)\right)\right)\right), \tag{13}$$

where the input x represents the time series input.

In our work, given the sequential multi-observation input $\mathbf{o}_t$ consisting of vehicle trajectories $\mathcal{P}_t \in \mathbb{R}^{T_h \times 3}$ and the GNSS measurements sequence $\mathcal{M}_t \in \mathbb{R}^{T_h \times N_{max} \times 4}$, we first employ two different embedding layers to map to the transformer latent space of dimension d_m and obtain two embeddings $\mathbf{E}_t^{pos} \in \mathbb{R}^{T_h \times d_m}$ and $\mathbf{E}_t^{gnss} \in \mathbb{R}^{T_h \times d_m}$. Then, the two embeddings are fed into the transformer for extracting temporal features. The temporal representations $\mathbf{z}_t^{pos}$ and $\mathbf{z}_t^{gnss}$ through the transformer can be denoted as:

$$\mathbf{z}_t^{pos} = \mathrm{Trans}_t\left(\mathbf{E}_t^{pos}\right), \tag{14}$$

$$\mathbf{z}_t^{gnss} = \mathrm{Trans}_t\left(\mathbf{E}_t^{gnss}\right). \tag{15}$$

2) GCN for extracting spatial features: To capture the inter-satellite relationships and extract spatial features from multi-constellation satellite measurements at the current timestep *t*, we adopt a GCN architecture following previous research (Zhao et al., 2023b). Notably, we omitted to extract spatial features of vehicle trajectories $\mathcal{P}_t$ since $\mathcal{P}_t$carries spatial information themselves. GCN usually takes a connected graph

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as input, where $\mathcal{V}$ represents the nodes and $\mathcal{E}$ represents the edges (Kipf and Welling, 2016). Thus, we first introduce the construction of a multi-constellation graph, including the representation of nodes and edges.

We denote the node set $\mathcal{V}_t = \{v_t^j | j = 1, \ldots, N_{max}\}$ as all visible satellites at the current timestep *t*, including BDS, GPS, Galileo, and GLONASS systems. Every node $\{v_t^j\}_{j=1,\ldots,N_{max}}$ contains a feature vector $\mathbf{m}_t^j \in \mathbb{R}^4$, which is composed of the LOS vector and the pseudorange residuals, forming the node feature matrix $\mathbf{M}_t \in \mathbb{R}^{N_{max} \times 4}$ mentioned above. Then, we denote the edge set $\mathcal{E}_t = \{(v_t^i, v_t^j) | i, j = 1, \ldots, N_{max}\}$. We infer the neighbor relationship between the satellite pairs (v_t^i, v_t^j) by evaluating the similarity between two satellites based on their constellation type and pseudorange residual values. We define the edge ε_t^{ij} between the satellite pairs (v_t^i, v_t^j) as follows,

$$\varepsilon_t^{ij} = \begin{cases} 1, if\left(C_t^i = C_t^j\right) or \left(i \neq j \; and \; \left|\rho_t^i - \rho_t^j\right| < \eta\right) \\ 0, else \end{cases} \tag{16}$$

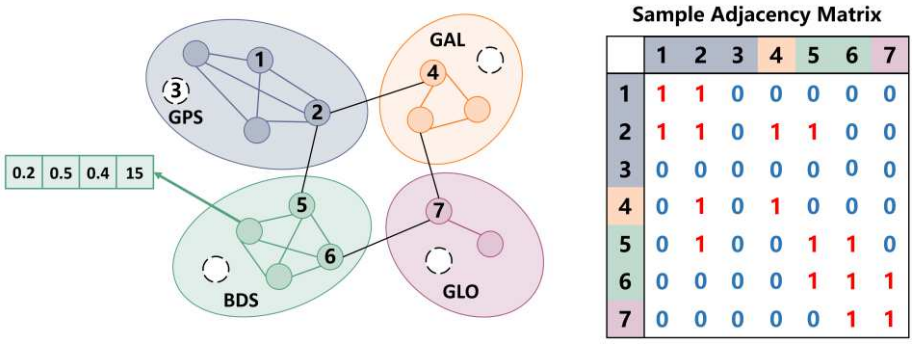
where C_t^i represents the constellation of satellite *i* and η is a hyper-parameter. Furthermore, we define an adjacency matrix $\mathbf{A}_t \in \mathbb{R}^{N_{max} \times N_{max}}$ to reflect connectivity between nodes. Therefore, we construct a multi-constellation graph $\mathcal{G}_t$ with node feature matrix $\mathbf{M}_t$ and adjacency matrix $\mathbf{A}_t$ to represent the spatial information between different satellites as follows,

$$\mathcal{G}_t = (\mathcal{V}_t, \mathcal{E}_t, \mathbf{A}_t, \mathbf{M}_t) \, . \tag{17}$$

Fig. 5 illustrates the constructed multi-constellation graph. Given a graph $\mathcal{G}_t$, GCN outputs the hidden features from the following update formula:

$$\mathbf{H}_t^{(l+1)} = \sigma\left(\widetilde{\mathbf{D}}^{-1/2} \widetilde{\mathbf{A}}_t \widetilde{\mathbf{D}}^{-1/2} \mathbf{H}_t^l \mathbf{W}_t^l\right), \tag{18}$$

where $\sigma(\cdot)$ is the ELU nonlinear activation function and $\mathbf{H}_t^{(l+1)}$ represents the hidden features of the (*l*+*1*)-th layer of GCN. $\widetilde{\mathbf{A}}_t = \mathbf{A}_t + I$ is the adjacency matrix after adding the self-loop. $\widetilde{\mathbf{D}}$ is the diagonal degree matrix of $\mathbf{A}_t$ and $\mathbf{W}_t^l$ refers to the trainable weights. In our framework, we implement an *L*-layer GCN to extract spatial features $\widehat{\mathbf{H}}_t^L$ from the multi-constellation satellite graph.



	1	2	3	4	5	6	7
1	1	1	0	0	0	0	0
2	1	1	0	1	1	0	0
3	0	0	0	0	0	0	0
4	0	1	0	1	0	0	0
5	0	1	0	0	1	1	0
6	0	0	0	0	1	1	1
7	0	0	0	0	0	1	1

Fig. 5 An illustration of constructed multi-constellation graph in the positioning correction environment. The dotted line circle represents the non-visible satellites.

3) Spatial-Temporal features Fusion and Multi-observation features Fusion: Based on the transformer and GCN modules proposed above, our MOGT derives temporal and spatial environmental features from multi-observation. Then we use a simple summation for fusion of spatial-temporal features $\mathbf{F}_t^{gnss} = \mathrm{Sum}\left(\mathbf{z}_t^{gnss}, \widehat{\mathbf{H}}_t^L\right)$. Moreover, to get a more comprehensive environmental representation, we employ a cross-attention strategy to model the interactions among the vehicle trajectory and GNSS measurements. In the cross-attention, the vehicle trajectory feature serves as the query, and the GNSS measurements feature serves as both the key and value. Ultimately, the multi-observation spatial-temporal representation can be represented as follows:

$$\mathbf{e}_t = \mathrm{MSA}\left(\mathbf{z}_t^{pos}, \mathbf{F}_t^{gnss}, \mathbf{F}_t^{gnss}\right), \tag{19}$$

Finally, the fusion feature $\mathbf{e}_t$ captures the impact of the spatial-temporal environmental feature of GNSS measurements and vehicle movement dynamics on GNSS positioning.

Self-supervised learning for encoding robust representation: To tackle the problem of performance degradation caused by data distribution shift in dynamically changing environments, we propose a self-supervised pretext task (SST) to train the encoder MOGT to encode the robust representation. Inspired by (Hansen and Wang, 2021), a pretext task is designed aimed at training the encoder MOGT to learn invariant representation from different views of the same observation, thereby ignoring the impact of dynamic changes. Through this pretext task, the encoder can learn environmental representation that is robust to the dynamic change of environments. Notably, in contrast to DRL-SSLDA (Li et al., 2024) that considers single-step

observation, our method adopts sequential observation to encode sequential representation, which helps the encoder capture dynamic patterns of stochastic noise in the driving scene. The details of SST are shown as follows:

Given the multi-observation sequence $\mathbf{o}_t$, we use a data augmentation method to model the stochastic noise caused by the dynamic change of environments in the driving scene and generate augmented data $\mathbf{o}'_t$. $\mathbf{o}_t$ and $\mathbf{o}'_t$ can be viewed as different views of the same observation. The non-augmented data $\mathbf{o}_t$ and augmented data $\mathbf{o}'_t$ are encoded through two identical encoders f_ψ and f_θ, followed by two identical projection heads g_ψ and g_θ to generate two feature vectors:

$$\mathbf{z}_t^\star = g_\psi\left(f_\psi(\mathbf{o}_t)\right), \tag{20}$$

$$\mathbf{z}'_t = g_\theta\left(f_\theta(\mathbf{o}_t)\right). \tag{21}$$

Then we use a predictor h_θ to learn a mapping $\hat{\mathbf{z}}_t = h_\theta(\mathbf{z}'_t)$ to match the target feature $\mathbf{z}_t^\star$. A consistency loss is used to compute the similarity between $\hat{\mathbf{z}}_t$ and $\mathbf{z}_t^\star$, defined as

$$\mathcal{L}_{SST}(\hat{\mathbf{z}}_t, \mathbf{z}_t^\star) = \left\| \frac{\hat{\mathbf{z}}_t}{\|\hat{\mathbf{z}}_t\|_2} - \frac{\mathbf{z}_t^\star}{\|\mathbf{z}_t^\star\|_2} \right\|_2. \tag{22}$$

The consistency loss $\mathcal{L}_{SST}$ is optimized jointly with the RL objective. It is significant to note that the parameter ψ is updated by the exponentially moving average (EMA) of the parameter θ at time *t*:

$$\psi_{t+1} \leftarrow \alpha\psi_t + (1-\alpha)\theta_t. \tag{23}$$

where $\alpha \in (0,1]$ is a momentum coefficient. In our experiments, we set $\alpha = 0.99$ to achieve optimal performance.

Continuous position-corrected action space setting : Our goal is to correct the initial position to the accurate position. Therefore, we define the action as a continuous position-corrected operation to the initial position, denoted as $\Delta\mathbf{p}_t = (\Delta x_t, \Delta y_t, \Delta z_t)$. Compared to the setting of a discrete action space, the continuous action space setting can effectively avoid the learning complexity caused by a large number of discrete actions. The position-corrected operation is sampled from a Gaussian distribution determined by the action $\mathbf{a}_t$ that the actor outputs. The procedure is described as follows:

1) The action $\mathbf{a}_t$ from the output of the actor is defined as $\mathbf{a}_t = \left(\mu_x, \sigma_x, \mu_y, \sigma_y, \mu_z, \sigma_z\right)$, which determines the mean and standard deviation of the

Gaussian distribution.

2) The position-corrected operation is sampled from the Gaussian distribution based on $\mathbf{a}_t$, *i.e.*, $\Delta x_t \sim \mathcal{N}(\mu_x, \sigma_x)$, $\Delta y_t \sim \mathcal{N}(\mu_y, \sigma_y)$, $\Delta z_t \sim \mathcal{N}(\mu_z, \sigma_z)$.

3) Thus, the initial position can be corrected as follows:

$$\widehat{\mathbf{p}}_t = \mathbf{p}_t + \Delta\mathbf{p}_t = (x_t + \Delta x_t, y_t + \Delta y_t, z_t + \Delta z_t), \tag{24}$$

where $\widehat{\mathbf{p}}_t$ is the corrected position.

Reward function setting : To ensure that the reward effectively enhances the learning of the correction policy, we employ the correction advantage error as the reward function:

$$r_t = \|\mathbf{p}_t - \mathbf{p}_t^{true}\|_2 - \|\widehat{\mathbf{p}}_t - \mathbf{p}_t^{true}\|_2, \tag{25}$$

where $\mathbf{p}_t^{true}$ is the ground truth position obtained by map matching algorithms (Quddus et al., 2007) or accurate centimeter-level positioning systems (Fu et al., 2020).

Actor-critic network using PPO for policy learning

In our proposed framework, we employ two DNN modules to construct the actor and critic components. Given the latent representation $\mathbf{e}_t$ at timestep *t*, the actor network π_{θ_a} outputs a continuous action for position correction, and the critic V_{θ_c} estimates the observation value, which can be denoted as:

$$\mathbf{a}_t = \pi_{\theta_\mathrm{a}}(\mathbf{e}_t) = \pi_{\theta_\mathrm{a}}\big(f_\theta(\mathbf{o}_t)\big), \tag{26}$$

$$V^{\pi}(\mathbf{o}_t) \approx V_{\theta_\mathrm{c}}(\mathbf{e}_t) = V_{\theta_\mathrm{c}}\big(f_\theta(\mathbf{o}_t)\big), \tag{27}$$

To ensure the stability of actor-critic network training, we implement the benchmark algorithm PPO to learn the optimal policy (Schulman et al., 2017). The actor network is trained using the clipped surrogate objective of PPO to ensure training stability, which is defined as:

$$\mathcal{L}_\mathrm{a}^{CLIP}(\theta_\mathrm{a}) = \mathbb{E}_t\big[\min\big(r_t(\theta_\mathrm{a})\hat{A}_t, \mathrm{clip}(r_t(\theta_\mathrm{a}), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\big], \tag{28}$$

where $r_t(\theta_\mathrm{a}) = \pi_{\theta_\mathrm{a}}(\mathbf{a}_t|\mathbf{o}_t)/\pi_{\theta_\mathrm{a}^{old}}(\mathbf{a}_t|\mathbf{o}_t)$ denotes the ratio of the new policy to the old policy, $\hat{A}_t$ denotes the generalized advantage estimation (Schulman et al., 2015) at timestep *t*, defined as,

$$\hat{A}_t(\mathbf{e}_t, \mathbf{a}_t) = \sum_{i=t}^{T-1}(\gamma\rho)^{i-t}\delta_i, \tag{29}$$

where $\delta_i = r_t + \gamma V_{\theta_\mathrm{c}}(\mathbf{e}_{t+1}) - V_{\theta_\mathrm{c}}(\mathbf{e}_t)$. Furthermore, to achieve accurate observation value and long-term positioning accuracy, the critic network V_{θ_c} is trained using the mean-squared return error (MSRE) (Le et al., 2017), defined as,

$$\mathcal{L}_{\mathrm{c}}(\theta_{\mathrm{c}}) = \mathbb{E}_t\left[\frac{1}{2}\left\|\sum_{i=t}^{T}\gamma^{i-t}r_i - V_{\theta_{\mathrm{c}}}(\mathbf{e}_t)\right\|_2^2\right], \tag{30}$$

where the first term $\sum_{i=t}^{T}\gamma^{i-t}r_i$ denotes the discounted cumulative return with T steps vehicle trajectory.

Deep Reinforcement Learning with Robust Spatial-Temporal Representation for Improving GNSS Positioning Correction

In this section, we present our overall DRL-RSTR method for improving GNSS positioning correction.

1) Firstly, we process the GNSS measurements received by the receiver, and then we use a conventional model-based positioning method, the KF, to obtain the initial positions.

2) The sequential multi-observation $\mathbf{o}_t$ is set as the model input, including a GNSS measurement sequence and a vehicle trajectory. Furthermore, the latent representation $\mathbf{e}_t$ is obtained through our MOGT. The actor network outputs the action $\mathbf{a}_t$, and the critic network outputs the estimation $V_{\theta_{\mathrm{c}}}(\mathbf{e}_t)$ of the observation based on the latent representation $\mathbf{e}_t$.

3) Based on the action $\mathbf{a}_t$, the corrected position $\hat{\mathbf{p}}_t$ can be obtained, and the environment returns a reward $r_t(\mathbf{o}_t, \mathbf{a}_t)$ and the next observation $\mathbf{o}_{t+1}$. Then the experience tuple $(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1})$ is stored in the replay buffer $\mathcal{D}$.

4) The parameters θ can be updated using PPO by calculating losses $\mathcal{L}_{\mathrm{a}}^{CLIP}(\theta_{\mathrm{a}})$ and $\mathcal{L}_{\mathrm{c}}(\theta_{\mathrm{c}})$.

5) When the iteration number reaches the SST update frequency, the parameters of MOGT $\theta_{\mathrm{trans}}, \theta_{\mathrm{gcn}}$ and θ_{ca} can be updated by minimizing the loss $\mathcal{L}_{SST}$. This improves the robustness of representation through MOGT in dynamically changing environments.

Finally, the proposed DRL-RSTR algorithm for model training is summarized in Algorithm 1.

Algorithm 1: DRL with Robust Spatial-Temporal Representation for Improving GNSS Positioning Correction

Input: Capacity of replay buffer *N*, reward discount factor γ, batch size *B*, SST update frequency f, learning rate lr, iteration number num.
Output: Trained DRL-RSTR model with parameter set $\theta = \{\theta_{\mathrm{trans}}, \theta_{\mathrm{gcn}}, \theta_{\mathrm{ca}}, \theta_{\mathrm{a}}, \theta_c\}$.
1: Initialization: DRL-RSTR model parameter set $\theta = \{\theta_{\mathrm{trans}}, \theta_{\mathrm{gcn}}, \theta_{\mathrm{ca}}, \theta_{\mathrm{a}}, \theta_c\}$, replay buffer $\mathcal{D}$ of capacity *N,* iteration step i = 1;

2: **for** $iter = 1, \ldots, num$ **do**
3: Collect experiences tuples $(\mathbf{o}_t, \mathbf{a}_t, r_t, \mathbf{o}_{t+1})$ with the policy π_{θ_a} by interacting with t he environment and store them in $\mathcal{D}$;
4: Calculate actor loss function $\mathcal{L}_a^{CLIP}(\theta_a)$ with the experiences stored in $\mathcal{D}$;
5: Calculate critic loss function $\mathcal{L}_c(\theta_c)$ with the experiences stored in $\mathcal{D}$;
6: Update the DRL-RSTR model parameter set $\theta = \{\theta_{trans}, \theta_{gcn}, \theta_{ca}, \theta_a, \theta_c\}$ with lr base d on the Adam method by computing gradients from loss functions $\mathcal{L}_a^{CLIP}(\theta_a)$ and $\mathcal{L}_c(\theta_c)$;
7: **if** $iter \; mod \; f == 0$ **then**
8: Generate augmented multi-observation $\mathbf{o}_t'$;
9: Calculate SST loss function $\mathcal{L}_{SST}$ with original multi-observation $\mathbf{o}_t$ and augmente d multi-observation $\mathbf{o}_t'$;
10: Update the feature encoder MOGT parameter $\theta_{trans}, \theta_{gcn}$ and θ_{ca} by minimizing $\mathcal{L}_{SST}$ based on the Adam method;
11: end for
12:end for

Experiments

To analyze both the effectiveness and limitations of our method comprehensively, we evaluate the whole pipeline with two major datasets, including the self-collected GZGNSS dataset in China and the public GSDC dataset collected in America. Our experiments are implemented on NVIDIA's GeForce RTX 3090 graphics card.

Datasets

1) **GZGNSS dataset** (Tang et al., 2024) was collected using N307-5D GNSS receivers in Guangzhou, China. This dataset consists of multi-constellation measurements from Beidou, GPS, Galileo and QZSS systems. The GZGNSS dataset contains 116 vehicle trajectories from different scenes. Similarly, this dataset is categorized into four scenario subsets for experimental validation: open road, urban forest, urban canyon, and overpass. We only choose the trajectories in urban forest, urban canyon, and overpass for our experiment. As illustrated in Fig. 6, the trajectory example in urban forest is surrounded by tall woods, the trajectory example in urban canyon is surrounded by tall buildings, and overpass trajectory example is under the overpass. Furthermore, we analyze the positioning error distribution of different scenarios in Fig. 7. It shows that the positioning errors of trajectory in overpass and urban forest are mostly less than 5 meters. However, in urban canyon, some positioning errors exceed 10 meters, and even reach 50 meters. It indicates that there are also non-negligible differences in the positioning error distribution of different scenarios. To validate the generalization of

our DRL-RSTR, we use the trajectories in single scenario for model training, the remaining unseen trajectories in other scenarios for model testing.

Fig. 6 Trajectory examples in urban environments from the GZGNSS dataset.

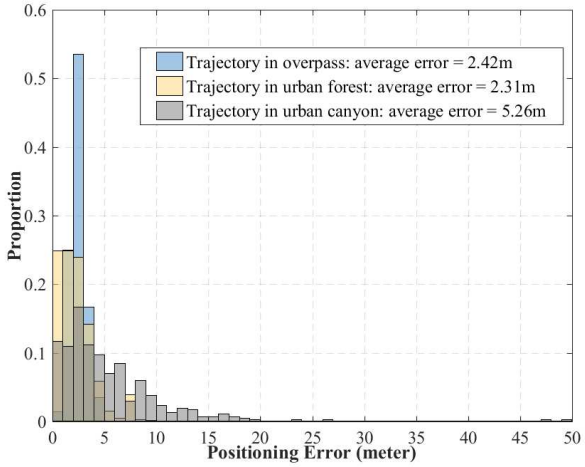


Fig. 7 Positioning error distribution of the complex urban environment from the GZGNSS dataset.

2) **GSDC dataset** (Fu et al., 2020) consists of GNSS measurement from various driving trajectories, which is collected by different Android smartphones in San Francisco and Los Angeles International Airport in America. This dataset contains 79 vehicle trajectories, divided into two subsets based on collection environments: 32 trajectories collected in complex urban environments and 47 trajectories collected in highway environments. In our work, we only capture the trajectories of the urban environments for the experiment. As illustrated in Fig. 8, the urban trajectory example involves dense buildings or overpasses, resulting in severe positioning deviations. We further analyze the positioning error distribution of several trajectories in Fig. 9. It shows that the data distribution of different vehicle trajectories is quite different, indicating that there may be a data distribution shift between training and testing phases. In our experiment, we employ the model-based SPP positioning method KF to obtain the initial positions $\mathbf{p}_t$. To enable the model to learn accurate positioning correction, the ground truth positions $\mathbf{p}_t^{true}$ are obtained by the GNSS-INS integrated NovAtel SPAN system (Fu et al.,

2020), which can provide the centimeter-level positioning accuracy. In the experiment, we use 50% of the trajectories of one subset for training, the remaining 50% of the unseen trajectories for testing.

Fig. 8 Trajectory examples in urban environments from the GSDC dataset. "2021-04-21-US-MTV-2" and "2021-12-08-US-LAX-5" represent the codes of trajectories.

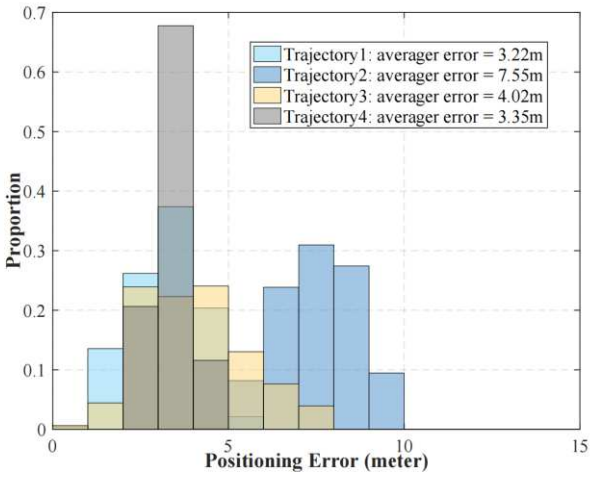


Fig. 9 Positioning error distribution of the complex urban environment from the GSDC dataset.

Comparison Methods

To evaluate the proposed approaches comprehensively, we compare our DRL-RSTR with the existing model-based method, state-of-the-art (SOTA) DRL-based methods:

1) **Model-based method**: We use model-based method KF as baseline. Our DRL-RSTR correct positions based on the initial positions obtained by KF.

2) **DRL-based methods**: Five SOTA DRL-based methods are chosen for comparison, i.e., A3C (Zhang and Masoud, 2020), Multi-LSTMPPO (Zhao et al., 2023a), LSTMPPO-ARAM (Tang et al., 2024), DRL-SSLDA (Li et al., 2024), and GNNRRL (Zhao et al., 2023b). A3C only employs the vehicle trajectories as input and a discrete action space for positioning correction, Multi-LSTMPPO uses multi-observation as input and a continuous action space with PPO for position-corrected

policy learning. LSTMPPO-ARAM introduce an adaptive reward augmentation method to address the performance degradation caused by inadequate real-time training data on the basis of Multi-LSTMPPO, DRL-SSLDA employ data augmentation and self-supervised learning to increase training data to address performance degradation in unseen urban scenarios. LSTMPPO-ARAM and DRL-SSLDA can effectively enhance the generalization of model by increasing the training data. GNNRRL consider a multi-constellation graph as input to extract the spatial feature in the current environment.

Implementation Details

For sequential multi-observation setting, the historical horizon is set to T_h and the number of visible satellites is fixedly set to N_{max}. For all DRL-based methods, the reward discount factor is set to $\gamma = 0.99$, with a replay buffer capacity of $N = 128$ and a batch size of $B = 128$. In our proposed SST framework, the encoder f_θ is implemented using the MOGT module, the projector g_θ is a multi-layer perceptron (MLP) with batch normalization (Ioffe and Szegedy, 2015), and the predictor h_θ is a linear layer with 256 neurons. The loss functions $\mathcal{L}_{\mathrm{a}}^{CLIP}(\theta_{\mathrm{a}})$, $\mathcal{L}_{\mathrm{c}}(\theta_{\mathrm{c}})$ and $\mathcal{L}_{SST}$ are all optimized using the Adam optimizer (Kingma, 2014). The update frequency of the SST module is set to $f = 2$. The learning rate lr used for experiments on the GZGNSS and GSDC datasets is further analyzed and discussed in the ablation study.

Experiment on the GZGNSS dataset

For the GZGNSS dataset, to validate the generalization of our method, we select the urban canyon and overpass environments respectively as training scenarios, and evaluate the performance in the remaining two environments. As can be seen in Table 1 and Table 2, we construct four different settings including "$\mathbf{C} \rightarrow \mathbf{O}$", "$\mathbf{C} \rightarrow \mathbf{F}$", "$\mathbf{O} \rightarrow \mathbf{C}$", and "$\mathbf{O} \rightarrow \mathbf{F}$". The notion "$\mathbf{C} \rightarrow \mathbf{O}$" indicates training in the urban canyon ($\mathbf{C}$) environment and testing in the overpass ($\mathbf{O}$) environment. Similarly, the notion "$\mathbf{C} \rightarrow \mathbf{F}$" indicates training in the urban canyon ($\mathbf{C}$) environment and testing in the urban forest ($\mathbf{F}$) environment.

Overall, our DRL-RSTR obtains the lowest horizontal error across all four settings. Compared to the baseline KF, DRL-RSTR trained in the $\mathbf{C}$ environment improves

positioning accuracy by 51.2% while training in the **O** environment improves positioning accuracy by 40.9%. The results demonstrate the superior positioning correction and generalization performance of DRL-RSTR across different urban scenarios. DRL-RSTR in the setting "**O → F**" obtain better positioning performance than the setting "**C → F**". We speculate that DRL-RSTR trained in a more complex overpass environment can enhance its ability to generalize to other urban environments. Furthermore, compared with DRL-SSLDA that also adopts self-supervised learning task, our method shows better positioning accuracy. The reason might be that DRL-SSLDA only consider single-step invariant representation and DRL-RSTR consider sequential invariant representation which is more adaptable to dynamic changing environments. Additionally, we demonstrate the positioning performance of different methods in various urban scenarios in Fig. 10, including ground truth, KF and our DRL-RSTR. As can be seen, the positioning offsets in urban canyon and overpass environments are more severe than in urban forest, indicating stronger signal interference in the former two scenarios. Moreover, the red dotted line (DRL-RSTR) more closely aligns with the blue solid line (ground truth) than the green dotted line (KF), indicating that DRL-RSTR effectively corrects the initial positions obtained by KF in the GZGNSS dataset.

Table 1 Comparison of average error with the SOTA GNSS positioning methods trained in the urban canyon environment on GZGNSS dataset. The best performance is highlighted in **bold**. The second-best results is marked with an underline. "↑" denotes higher is better. "↓" denotes lower is better.

Trained on the trajectories in the urban canyon environment								
Method		Temporal	Spatial	SST	**C → O**	**C → F**	Average error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			6.18 ± 2.52	6.19 ± 1.71	6.19 ± 2.12	-
DRL-based	A3C				3.61 ± 2.54	3.14 ± 1.62	3.38 ± 2.08	45.4
	Multi-LSTMPPO	✓			3.99 ± 2.67	3.80 ± 1.73	3.90 ± 2.20	37.0
	LSTMPPO-ARAM	✓			4.19 ± 2.61	3.89 ± 1.75	4.04 ± 2.18	34.7
	DRL-SSLDA	✓		✓	~~3.41 ± 2.61~~	2.99 ± 1.52	3.20 ± 2.07	48.3
	GNNRRL		✓		3.46 ± 2.63	~~2.92 ± 1.53~~	~~3.19 ± 2.09~~	~~48.5~~
	DRL-RSTR (Ours)	✓	✓	✓	$\mathbf{3.32 \pm 2.61}$	$\mathbf{2.72 \pm 1.51}$	$\mathbf{3.02 \pm 2.06}$	**51.2**

Table 2 Comparison of average error with the SOTA GNSS positioning methods trained in the overpass environment on GZGNSS dataset.

Trained on the trajectories in the overpass environment								
Method		Temporal	Spatial	SST	**O → C**	**O → F**	Average error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			6.37 ± 3.36	6.19 ± 1.71	6.28 ± 2.54	-
DRL-based	A3C				5.03 ± 3.53	3.09 ± 1.65	4.06 ± 2.59	35.4
	Multi-LSTMPPO	✓			~~4.88 ± 3.41~~	2.94 ± 1.56	3.91 ± 2.49	37.7
	LSTMPPO-ARAM	✓			4.95 ± 3.54	3.02 ± 1.50	3.99 ± 2.52	36.7
	DRL-SSLDA	✓		✓	4.95 ± 3.59	~~2.81 ± 1.61~~	~~3.88 ± 2.60~~	~~38.2~~
	GNNRRL		✓		5.03 ± 3.55	2.96 ± 1.53	4.00 ± 2.54	36.8
	DRL-RSTR (Ours)	✓	✓	✓	$\mathbf{4.79 \pm 3.54}$	$\mathbf{2.56 \pm 1.38}$	$\mathbf{3.68 \pm 2.46}$	**41.4**

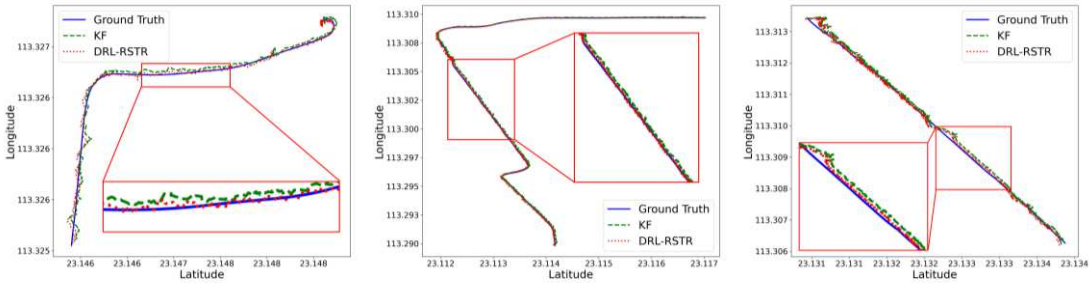


Fig. 10 The visualization of positioning performance of different methods in various urban scenarios.

Experiment on the GSDC dataset

In this section, we compare the performance of the proposed DRL-RSTR with other methods on the GSDC dataset. As shown in Table 3, DRL-RSTR thoroughly outperforms all the other positioning methods. Notably, while LSTMPPO-ARAM and DRL-SSLDA enhance generalization through data augmentation, A3C, Multi-LSTMPPO, and GNNRRL exhibit significantly inferior performance due to their limited generalization. Compared with DRL-SSLDA, our DRL-RSTR has improved the performance by 2%. The reason for the improvement is that our SST utilizes sequence input to learn sequential invariant representation, which considers the dynamic patterns of the urban environments. Additionally, by comparing to the baseline KF, DRL-RSTR can improve the positioning accuracy by 6.5%, indicating that our method can effectively correct the initial positions from KF. Fig. 11 displays the average error of trajectories collected by different smartphones in different areas. It can be seen that the error of these trajectories changes drastically due to the dynamic changes of the surrounding environment of the vehicle in complex urban areas. In the right figure of

Fig. 11b, it is obvious that the error suddenly changed from 2m to 10m about 700 steps. Even so, our method is still able to follow the changes and correct the errors effectively.

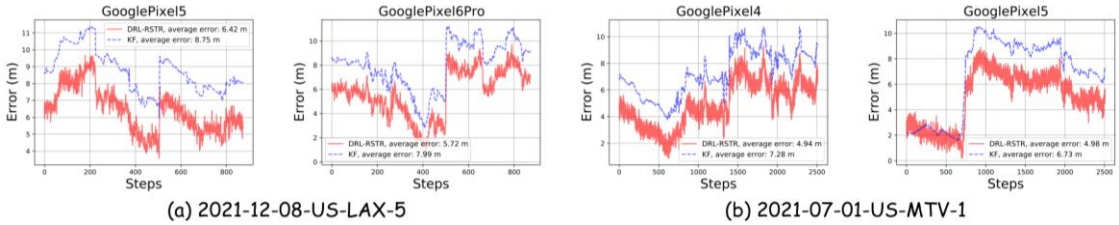


Fig. 11 Evaluation performance of the error of trajectories collected by different smartphones in LAX and MTV from the GSDC dataset.

Table 3 Comparison of average error with the SOTA GNSS positioning methods on GSDC dataset.

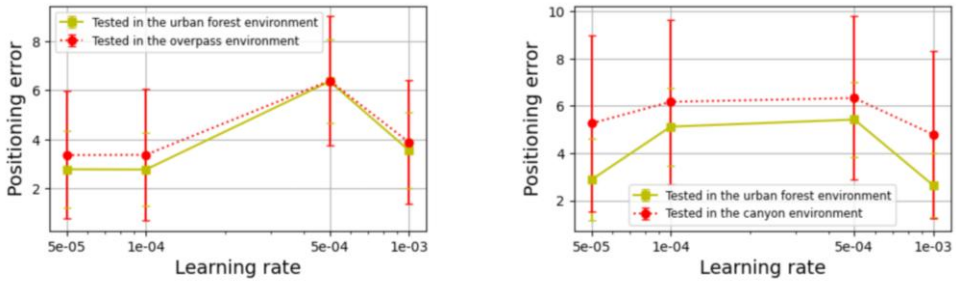
Tested on the unseen urban trajectories of the other half of the GSDC dataset						
Method		Temporal	Spatial	SST	Positioning error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			5.54 ± 1.64	-
DRL-based	A3C				5.33 ± 1.67	3.9
	Multi-LSTMPPO	✓			5.41 ± 1.76	2.3
	LSTMPPO-ARAM	✓			5.44 ± 1.72	1.8
	DRL-SSLDA	✓		✓	~~5.29 ± 1.81~~	~~4.5~~
	GNNRRL		✓		5.42 ± 1.76	2.2
	DRL-RSTR (Ours)	✓	✓	✓	$\mathbf{5.18 \pm 1.78}$	**6.5**

Ablation Study

In this section, we first analyze the performance of different learning rates lr for the GZGNSS and GSDC datasets. Then we verify the effectiveness of the spatial-temporal representation and the SST module by comparing the performances of DRL-RSTR (w/o Temporal), DRL-RSTR (w/o Spatial), DRL-RSTR (w/o SST) and the complete DRL-RSTR model.

The performance of the proposed DRL-RSTR with a range of learning rate lr within the range $[5e^{-5}, 1e^{-3}]$ is shown in Fig. 12. For the GZGNSS dataset, we select $lr = 5e^{-5}$ for the model trained in the urban canyon environment and $lr = 1e^{-3}$ for the model trained in the overpass environment. For the GSDC dataset, as shown in Fig. 13, we select $lr = 1e^{-3}$ for the model trained on half of the urban trajectories on GSDC dataset. As shown in Tabels 4, 5 and 6, compared to DRL-RSTR (w/o Temporal) and DRL-RSTR (w/o Spatial) that consider only spatial or temporal aspects of GNSS observations individually, DRL-RSTR obtains higher positioning accuracy, which

demonstrates the effectiveness of jointly modeling both spatial and temporal correlations of GNSS observations. Additionally, compared to DRL-RSTR (w/o SST), the complete DRL-RSTR model also shows significant improvements in performance and generalization, demonstrating the effectiveness of the SST module in learning robust representation in dynamic changing urban environments. As can be seen in Table 5 and Table 6, the performance of DRL-RSTR (w/o SST) trained in **O** environment significantly deteriorates in unseen scenarios, primarily due to its lack of robustness against environmental dynamic changes. Therefore, the results of the ablation study demonstrate the integration of spatial-temporal representation learning and the SST module is essential for achieving robust and accurate GNSS positioning in dynamic changing urban environments.



(a) Trained in the urban canyon environment (b) Trained in the overpass environment

Fig. 12 Performance of the proposed DRL-RSTR with a range of learning rate in different positioning correction environments on GZGNSS dataset.

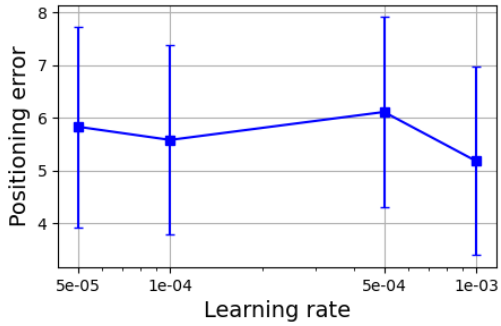


Fig. 13 Performance of the proposed DRL-RSTR with a range of learning rate on GSDC dataset.

Table 4 Comparison of average error of different methods trained in the urban canyon environment for ablation study on GZGNSS dataset.

Trained on the trajectories in the urban canyon environment								
Method		Temporal	Spatial	SST	**C → O**	**C → F**	Average error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			6.18 ± 2.52	6.19 ± 1.71	6.19 ± 2.12	-
DRL-based	DRL-RSTR		✓	✓	3.55 ± 3.39	3.19 ± 1.50	3.37 ± 2.45	45.6

	(w/o Temporal)							
	DRL-RSTR (w/o Spatial)	✓		✓	~~3.33 ± 2.60~~	2.84 ± 1.46	~~3.09 ± 2.03~~	~~50.1~~
	DRL-RSTR (w/o SST)	✓	✓		3.35 ± 2.60	~~2.82 ± 1.53~~	3.09 ± 2.07	50.1
	DRL-RSTR (Full version)	✓	✓	✓	**3.32 ± 2.61**	**2.72 ± 1.51**	**3.02 ± 2.06**	**51.2**

Table 5 Comparison of average error of different methods trained in the overpass environment for ablation study on GZGNSS dataset.

Trained on the trajectories in the overpass environment								
Method		Temporal	Spatial	SST	**O → C**	**O → F**	Average error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			6.37 ± 3.36	6.19 ± 1.71	6.19 ± 2.12	-
DRL-based	DRL-RSTR (w/o Temporal)		✓	✓	5.28 ± 3.55	2.75 ± 1.34	4.02 ± 2.45	36.0
	DRL-RSTR (w/o Spatial)	✓		✓	~~4.79 ± 3.56~~	~~2.61 ± 1.36~~	~~3.70 ± 2.46~~	~~41.1~~
	DRL-RSTR (w/o SST)	✓	✓		5.13 ± 3.49	3.50 ± 1.65	4.32 ± 2.57	31.2
	DRL-RSTR (Full version)	✓	✓	✓	**4.79 ± 3.54**	**2.56 ± 1.38**	**3.68 ± 2.46**	**41.4**

Table 6 Comparison of average error of different methods for ablation study on GSDC dataset.

Trained on the trajectories in the overpass environment						
Method		Temporal	Spatial	SST	Positioning error (m) ↓	Improvement (%) ↑
Model-based	KF (baseline)	✓			5.54 ± 1.64	-
DRL-based	DRL-RSTR (w/o Temporal)		✓	✓	~~5.27 ± 1.62~~	~~4.8~~
	DRL-RSTR (w/o Spatial)	✓		✓	5.27 ± 1.78	4.8
	DRL-RSTR (w/o SST)	✓	✓		5.50 ± 1.83	0.7
	DRL-RSTR (Full version)	✓	✓	✓	**5.18 ± 1.78**	**6.5**

Conclusion and future work

In this paper, we propose DRL-RSTR, a novel DRL-based method for GNSS positioning correction in dynamic changing urban environments. Our method fully consider the rich spatial-temporal features of GNSS observations and employs a self-supervised pretext task to enhance the robustness of the latent representation against

dynamic changes. First, we propose a multi-observation GCN-transformer to effectively encode the spatial-temporal representation from sequential GNSS observations. Second, a self-supervised pretext task is designed to improve the robustness of spatial-temporal representation, enabling the model to better adapt to dynamic changing urban environments. Finally, we validate our DRL-RSTR on the GZGNSS and GSDC datasets, which shows that our method outperform state-of-the-art model-based and learning-based methods in terms of positioning accuracy and generalization.

Although the proposed DRL-RSTR demonstrates notable improvements and strong generalization in dynamically changing urban environments, several limitations still exist. For instance, our method does not explicitly address GNSS outages scenarios, such as those encountered in tunnels. It limits the applicability of the model in situations where GNSS signals are partially or completely unavailable. In future works, we will consider additional sensor modalities, such as inertial measurement units (IMUs) or LiDAR to construct more comprehensive and reliable observations. By fusing these data sources, we aim at improving the positioning correction performance during GNSS outages.

Acknowledgment

This research is supported in part by National Natural Science Foundation of China under Grants 62273106, 62320106008, in part by Guangzhou Basic and Applied Basic Research Foundation SL2024A04J00551, and in part by GuangDong Basic and Applied Basic Research Foundation 2024A1515240064.

Data Availability

Researchers interested in accessing the datasets generated and analyzed in this study may contact the corresponding author with a reasonable request.

Declarations

Author contributions

Conceptualization: Zhenni Li, Peili Li, Shengli Xie; Methodology: Zhenni Li, Peili Li, Shengli Xie; Formal analysis and Investigation: Zhenni Li, Peili Li, Jianhao Tang;

Software: Zhenni Li, Peili Li, Jianhao Tang; Data Curation: Yulong Song, Jianhao Tang, Liji Chen; Writing original draft preparation: Zhenni Li, Peili Li; Writing review and editing: Zhenni Li, Peili Li, Jianhao Tang, Yiting Cai; Funding acquisition: Shengli Xie. All authors reviewed the manuscript.

Competing interests

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

References

Ahmad, F., Qiu, H., Eells, R., Bai, F., and Govindan, R. (2020). {CarMap}: Fast 3d feature map updates for automobiles. In 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20), pages 1063–1081.

Ahmed, S., Nielsen, I. E., Tripathi, A., Siddiqui, S., Ramachandran, R. P., and Rasool, G. (2023). Transformers in time-series analysis: A tutorial. Circuits, Systems, and Signal Processing, 42(12):7433–7466

Breßler, J., Reisdorf, P., Obst, M., and Wanielik, G. (2016). Gnss positioning in non-line-of-sight context—a survey. In 2016 IEEE 19th international conference on intelligent transportation systems (ITSC), pages 1147–1154. IEEE.

Chen, Y., Ren, K., Wang, Y., Fang, Y., Sun, W., and Li, D. (2023). Contiformer: Continuous-time transformer for irregular time series modeling. Advances in Neural Information Processing Systems, 36:47143–47175.

Eissfeller, B., Tiberius, C., Pany, T., Biberger, R., and Heinrichs, G. (2001). Real-time kinematic in the light of gps modernization and galileo. In Proceedings of the 14th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2001), pages 650–682

Fu, G. M., Khider, M., and Van Diggelen, F. (2020). Android raw gnss measurement datasets for precise positioning. In Proceedings of the 33rd international technical meeting of the satellite division of the Institute of Navigation (ION GNSS+ 2020),

pages 1925 – 1937.

Gao, X., Luo, H., Ning, B., Zhao, F., Bao, L., Gong, Y., Xiao, Y., and Jiang, J. (2020). Rl-akf: An adaptive kalman filter navigation algorithm based on reinforcement learning for ground vehicles. Remote Sensing, 12(11):1704.

Han, K., Lee, S., Song, Y.-J., Lee, H.-B., Park, D.-H., and Won, J.-H. (2021). Precise positioning with machine learning based kalman filter using gnss/imu measurements from android smartphone. In Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021), pages 3094 – 3102.

Hansen, N. and Wang, X. (2021). Generalization in reinforcement learning by soft data augmentation. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pages 13611 – 13617. IEEE.

Hsu, L.-T. (2018). Analysis and modeling gps nlos effect in highly urbanized area. GPS solutions, 22(1):7. Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International conference on machine learning,pages 448 – 456. pmlr.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems.

Kanhere, A. V., Gupta, S., Shetty, A., and Gao, G. (2022). Improving gnss positioning using neural-network-based corrections. NAVIGATION: Journal of the Institute of Navigation, 69(4).

Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

Le, L., Kumaraswamy, R., and White, M. (2017). Learning sparse representations in reinforcement learning with sparse coding. arXiv preprint arXiv:1707.08316.

Li, P., Li, Z., Hou, K., Tang, J., and Xie, S. (2024). Improving gnss positioning using deep reinforcement learning with self-supervised learning based data

augmentation. In Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024), pages 2645 – 2658.

Li, Z., Xu, G., Guo, J., and Zhao, Q. (2022). A sequential ambiguity selection strategy for partial ambiguity resolution during rtk positioning in urban areas. GPS solutions, 26(3):92.

Liu, S., Elangovan, V., and Xiang, W. (2019). A vehicular gps error prediction model based on data smoothing preprocessed lstm. In 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), pages 1–5. IEEE.

Liu, Z., Liu, J., Xu, X., and Wu, K. (2022). Deepgps: Deep learning enhanced gps positioning in urban canyons. IEEE Transactions on Mobile Computing, 23(1):376–392.

Medina, D., Li, H., Vila-Valls, J., and Closas, P. (2019). On robust statistics for gnss single point positioning. In 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pages 3281–3287. IEEE.

Mohanty, A. and Gao, G. (2023). Learning gnss positioning corrections for smartphones using graph convolution neural networks. NAVIGATION: Journal of the Institute of Navigation, 70(4).

Mohanty, A. and Gao, G. (2024). A survey of machine learning techniques for improving global navigation satellite systems. EURASIP Journal on Advances in Signal Processing, 2024(1):73.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2022). A time series is worth 64 words: Long-term forecasting with transformers. arXiv preprint arXiv:2211.14730.

Niu, Y., Xiong, C., and Li, Z. (2025). Gnss-rtk data denoising and displacement-based blind modal analysis of a long-span bridge. GPS Solutions, 29(1):1–23.

Quddus, M. A., Ochieng, W. Y., and Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. Transportation research part c: Emerging technologies, 15(5):312–328.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. arXiv preprint arXiv:1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Shin, H., Lee, J., and Sung, C.-k. (2018). Implementation of deep reinforcement learning on high precision gnss/ins augmentation system. In Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2018), pages 3179–3185.

Tang, J., Li, Z., Hou, K., Li, P., Zhao, H., Wang, Q., Liu, M., and Xie, S. (2024). Improving gnss positioning correction using deep reinforcement learning with an adaptive reward augmentation method. NAVIGATION: Journal of the Institute of Navigation, 71(4).

Tang, J., Li, Z., Yu, Q., Zhao, H., Zeng, K., Zhong, S., Wang, Q., Xie, K., Kuzin, V., and Xie, S. (2025). Deep reinforcement learning with robust augmented reward sequence prediction for improving gnss positioning. GPS Solutions,29(2):65.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Wang, J. (1999). Stochastic modeling for real-time kinematic gps/glonass positioning. Navigation, 46(4):297–305.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., and Sun, L. (2022). Transformers in time series: A survey. arXiv preprint arXiv:2202.07125.

Weng, X., Ling, K. V., and Liu, H. (2024). Prnet: A neural network for correcting pseudoranges to improve positioning with android raw gnss measurements. IEEE Internet of Things Journal.

Wu, N., Green, B., Ben, X., and O'Banion, S. (2020). Deep transformer models for time series forecasting: The influenza prevalence case. arXiv preprint arXiv:2001.08317.

Yue, B., Zhang, Z., Zhang, W., Luo, X., Zhang, G., Huang, H., Wu, X., Bao, K., and Peng, M. (2024). Design of an automatic navigation and operation system for a crawler-based orchard sprayer using gnss positioning. Agronomy,14(2):271.

Zhang, E. and Masoud, N. (2020). Increasing gps localization accuracy with reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 22(5):2615–2626.

Zhao, H., Li, Z., Chen, C., Wang, L., Xie, K., and Xie, S. (2023a). Fusing vehicle trajectories and gnss measurements to improve gnss positioning correction based on actor-critic learning. In Proceedings of the 2023 international technical meeting of the institute of navigation, pages 82–94.

Zhao, H., Li, Z., Wang, Q., Xie, K., Xie, S., Liu, M., and Chen, C. (2024). Improving performances of gnss positioning correction using multiview deep reinforcement learning with sparse representation. GPS Solutions, 28(3):98.

Zhao, H., Tang, J., Li, Z., Wu, Z., Xie, S., Wu, Z., Liu, M., and Kumara, B. T. (2023b). Efficient graph neural network driven recurrent reinforcement learning for gnss position correction. In Proceedings of the 36th International

Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2023), pages 216–230.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 11106–11115.

Zumberge, J. F., Heflin, M. B., Jefferson, D. C., Watkins, M. M., and Webb, F. H. (1997). Precise point positioning for the efficient and robust analysis of gps data from large networks. Journal of geophysical research: solid earth, 102(B3):5005–5017.

Author Biographies

Zhenni Li received the Ph.D degree in 2015 from School of Computer Science and Engineering, University of Aizu, Japan. She is currently a Professor in School of Automation, Guangdong University of Technology, China. Her research interests include machine learning, deep

reinforcement learning and high-precision GNSS positioning technology.

Peili Li is currently pursuing a B.S. degree at the Guangdong University of Technology , China. His current research interests mainly include deep reinforcement learning, transfer learning and GNSS positioning.

Jianhao Tang received the B.Sc. degree from the School of Automation, Guangdong University of Technology, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Automation, Guangdong University of Technology. His current research interests include deep reinforcement learning, dictionary learning and high-precision GNSS positioning technology.

Yulong Song received his Ph.D. degree in Information and Communication Engineering from Northwestern Polytechnical University in 2017. Currently, he is working as a senior engineer at Guangzhou Haige Communication Group Incorporated Company, located at No. 88, Nanyun 2nd Road, Guangzhou City, Guangdong Province, China. His current research focus is on the fields of satellite navigation baseband algorithms and SoC design.

Liji Chen received the B.Sc. degree from the School of Automation, Guangdong University of Technology, China, in 2022. He is currently pursuing the Ph.D. degree with the School of Automation, Guangdong University of Technology. His current research interests include satellite positioning technology, deep space exploration system and artificial intelligence.

Yiting Cai is currently pursuing a B.S. degree at the Guangdong University of Technology , China. His current research interests mainly include deep reinforcement learning and GNSS positioning.

Shengli Xie IEEE Fellow, Foreign Full Member (Academician) of the Russian Academy of Engineering, received the B.S. degree in mathematics from Jilin University, China in 1983, the M.S. degree in mathematics from Central China Normal University, China in 1995, and the Ph.D. degree in control theory and applications from South China University of Technology, China in 1997. He is currently a full professor, and the head of Key Laboratory of Intelligent Detection and The Internet of Things in Manufacturing (GDUT), Ministry of Education. His research interests include blind

868 signal processing, machine learning, and internet of things. He was awarded the Second Prize of
869 National Natural Science Award of China in 2009. He is an Associate Editor of IEEE TSMC.