GPS Solutions (2024) 28:98
https://doi.org/10.1007/s10291-024-01626-6

ORIGINAL ARTICLE

Improving performances of GNSS positioning correction using multiview deep reinforcement learning with sparse representation

Haoli Zhao[1,3] · Zhenni Li[1,3] · Qianming Wang[1,7] · Kan Xie[1,2] · Shengli Xie[4,5] · Ming Liu[6] · Ci Chen[1,2]

Received: 21 November 2023 / Accepted: 1 February 2024 / Published online: 5 April 2024

Abstract

High-accuracy GNSS positioning in urban environments is important for applications like safe autonomous driving, however, dynamic errors in complex urban environments limit positioning performances. Recently, deep learning-based (DL) approaches can obtain better GNSS positioning solutions in complex urban environments than model-based ones. However, DL-based approaches simply concentrate one-view GNSS observations as inputs, which are insufficient to model vehicle states accurately, and temporally continuous observations are highly correlated, leading to inaccurate positioning correction results. To solve the challenge, we propose a Sparse Representation-based Multiview Deep Reinforcement Learning model for positioning correction, which employs attention-based multiview fusion to process multiview observations, and uses sparse representation to alleviate disturbances from highly correlated observations. To represent the vehicle state sufficiently, we build a multiview positioning correction environment, and develop an attention-weighted multiview fusion module to fuse temporal features as belief states based on adaptively learned attention weights. To effectively process redundant and correlated multiview features, we impose the $\ell 1$ norm regularizer to learn sparse hidden representations and improve the precision of value estimation. Finally, we construct a sparse representation-driven multiview actor-critic positioning correction model to achieve high-accuracy GNSS positioning in complex urban environments. We validate performances in both Google Smartphone Decimeter Challenge (GSDC) datasets and our collected GNSS datasets in the Guangzhou area (GZGNSS). Experimental results show that our algorithm can improve localization performances with 27% improvements from WLS+KF in GSDC trajectories, 16% from RTK, and 6% from DL-based methods in GZGNSS trajectories.

Keywords Deep Reinforcement Learning · Global Navigation Satellite Systems Positioning · Sparse representation · Multiview learning

Introduction

With the development of the Global Navigation Satellite Systems (GNSS) in both hardware and software, the positioning accuracy of a Global Positioning System (GPS) device can reach centimeter levels in open places (Teunissen and Khodabandeh 2015). However, the accuracy of GNSS positioning is affected by different factors, including surrounding environments, devices, etc. Due to the interference of satellite signals by dynamic urban errors, e.g., multipath and Non-Line-Of-Sight (NLOS) errors, the positioning of the satellite navigation positioning system may reach tens of meters in different complex urban scenarios, such as urban canyons, overpasses, viaducts, urban forests, etc., which cannot meet the basic needs of continuous lane-level positioning of autonomous vehicles. At present, the satellite navigation positioning technology for intelligent driving still relies on

✉ Zhenni Li
lizhenni2012@gmail.com

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

[2] 111 Center for Intelligent Batch Manufacturing Based on IoT Technology(GDUT), Guangzhou 510006, China

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

[4] Guangdong Key Laboratory of IoT Information Technology (GDUT), Guangzhou 510006, China

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

[6] Thrust of Robotics and Autonomous Systems, Hong Kong University of Science and Technology, Guangzhou, China

[7] Techtotop Microelectronics Technology Co. Ltd., Guangzhou 510006, China

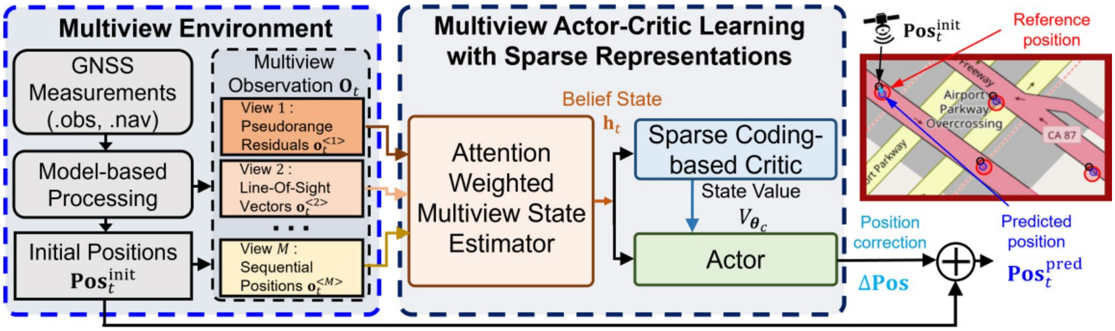

Multiview Environment
GNSS Measurements (.obs, .nav)
Model-based Processing
Initial Positions $\mathbf{Pos}_t^{\text{init}}$
Multiview Observation $\mathbf{O}_t$
View 1 : Pseudorange Residuals $\mathbf{o}_t^{<1>}$
View 2 : Line-Of-Sight Vectors $\mathbf{o}_t^{<2>}$
View M : Sequential Positions $\mathbf{o}_t^{<M>}$
Multiview Actor-Critic Learning with Sparse Representations
Belief State
$\mathbf{h}_t$
Attention Weighted Multiview State Estimator
Sparse Coding-based Critic
State Value
V_{θ_c}
Actor
$\mathbf{Pos}_t^{\text{init}}$
Reference position
Freeway
Airport Parkway Overcrossing
CA 87
rt Parkway
Position correction
$\Delta\mathbf{Pos}$
Predicted position
$\mathbf{Pos}_t^{\text{pred}}$

Fig. 1 A graphical architecture of the proposed MVDRL-SR algorithm for sequential positioning correction

model-based methods, and the high-precision positioning of satellite navigation in complex urban environments is still an open problem (Skog and Handel 2009).

Different from model-based methods, whose performances are restricted by prior model assumptions, learning-based methods can model complex urban environmental errors by training using data. Existing deep learning-based (DL) approaches employ different neural network models, e.g., Transformer (Kanhere et al. 2022) and Graph Convolutional Neural Networks (GCNN) (Mohanty and Gao 2022), to predict positioning correction values in complex urban environments for every time step. However, these works simply concentrate Pseudorange Residuals (PRR) and Line-Of-Sight (LOS) vectors as one-view GNSS inputs, ignoring relationships between different GNSS features, which are insufficient to model precise vehicle states, and these temporally continuous inputs are highly correlated, leading to disturbances in training and positioning correction. Moreover, DL models consider the localization problem of each position discretely, ignoring sequential relationships between positions. On the other hand, the Deep Reinforcement Learning-based (DRL) method (Zhang and Masoud 2020) uses highly correlated historically predicted positions as input to train a positioning correction policy based on rewards connecting adjacent states or observations, helping the agent consider temporal relationships. However, all these existing works suffered from insufficient and temporally continuous model inputs, leading to inaccurate vehicle modeling and interference in training.

Motivation

To address issues because of insufficient and temporally continuous model inputs in DRL methods, this paper proposes a Sparse Representation-based Multiview Deep Reinforcement Learning model (MVDRL-SR) for positioning correction. To model the vehicle agent more accurately, we construct a multiview positioning correction reinforcement learning (RL) environment with measure residuals, LOS vectors, and sequential vehicle positions, and employ the Long Short-Term Memory (LSTM) module to extract historical information from sequential observations of different views. Then, we effectively fuse the belief state based on adaptively learned attention weights, which can help the agent decide which view is more informative and valuable during training. To effectively process highly correlated features from temporally continuous multiview observations, we employ the $\ell 1$ norm regularizer in the critic to promote sparse hidden representation during network propagation, which can effectively reduce coherences of representations, increase the precision of value estimation, and thus improve the stability of the positioning correction policy. The diagram of MVDRL-SR is summarized in Fig. 1, including attention-weighted multiview fused belief states, and sparse coding-based sparse critic representation learning. In the end, this paper validates the proposed method in different real-world GNSS datasets, e.g., open Google Smartphone Decimeter Challenge (GSDC) datasets and our collected GNSS datasets in the Guangzhou area (GZGNSS), where the proposed MVDRL-SR can outperform both model-based methods with 27% from Kalman Filter (KF) processed Weight Least Squares (WLS) solutions (Verhagen and Teunissen 2017; Medina et al. 2019) in GSDC trajectories and 16% improvements from RTK in GZGNSS trajectories; and learning-based state-of-the-art methods with 6% from DL-based methods.

The main contributions of the paper are summarized as follows,

1. To effectively model the vehicle, we construct the multiview RL environment by complementing sequential position observations, which are exploited by LSTM modules for historical features and fused based on adaptively learned attention weights considering relationships between views.
2. To alleviate interference from redundant and correlated multiview temporally continuous observations, we employ the $\ell 1$ norm regularizer and corresponding

proximal operator in the critic to promote sparse hidden representation during network propagation and thus increase the precision of value estimation.
3. To achieve an accurate sequential positioning correction policy, we train the actor-critic DRL model with the cumulative value estimation and time-difference advantage estimation, whose performances are validated in both the open GSDC dataset, and our collected GZGNSS dataset.

To validate the performance of the proposed MVDRL-SR, we compared it with different model-based and learning-based algorithms in the experimental section, including,

1. **Model-based Algorithms:** In the GSDC dataset, we use the Single Point Positioning (SPP) method WLS+KF (Verhagen and Teunissen 2017; Medina et al. 2019) which employ carrier-smoothed pseudorange[1] and temporal information to obtain solutions as the model-based baselines. In our collected GZGNSS dataset, we employ the carrier-phase differential RTK solutions (Shu et al. 2017; Li et al. 2022) as the model-based baselines, which use GNSS measurements from a base station to enhance the rover performances with a partial ambiguity resolution.
2. **Deep Learning-based Algorithms:** Two state-of-the-art deep learning-based GNSS positioning correction methods are chosen for validation, i.e., SetTransformer (Kanhere et al. 2022) and GCNN (Mohanty and Gao 2022), which employ different neural network modules to learn to correct model-based solutions. SetTransformer employs the attention-based transformer module to learn from only the GPS-L1C constellation, and we set the correction output in ECEF axes. On the other side, GCNN employs the GIN module to learn from three constellations of GNSS measurements, i.e., GPS, GLONASS, and GALILEO.
3. **RL-based Algorithms:** Two state-of-the-art RL algorithms for positioning correction tasks are selected for comparison, i.e., A3C (Zhang and Masoud 2020) and Multi-LSTMPPO (Zhao et al. 2023). A3C only employs the vehicle trajectories as observations and simply resizes them into one vector to form the belief states. Moreover, this method uses a complex discrete action space that estimates values for 441 actions. Because the MSE reward setting used in this work is not functional in experiments, we use the correction advantage error proposed in this paper for this algorithm. Multi-LSTMPPO employs multi-inputs as observations, but simply concentrates different GNSS measurements and vehicle trajectories as model inputs, and does not consider the influence of temporally continuous correlated observations.

Related work

Recently, there have emerged different model-based approaches to enhance positioning accuracy in urban areas (Wen et al. 2020; Zhu et al. 2018). However, conventional model-based methods are limited by rigid prior assumptions on sensors, model parameters, etc., and can hardly adapt to dynamically changing multipath error models in urban scenes. Using high-precision maps and inertial navigation can help improve urban localization performances, but can also hardly match the requirements of continuous high-precision absolute positioning for autonomous driving in complex urban environments. Nevertheless, high production costs, strong hardware restrictions, and poor scene generalization make high-precision map-based approaches face limitations in applicable scenarios (Wang et al. 2021; Cai et al. 2018). Moreover, some works employ LiDAR measurement-based 3D model mapping to predict NLOS signals for enhancing positioning accuracy in urban navigation (Groves and Adjrad 2017; Xin et al. 2022; Liu et al. 2022), but also are limited by the high cost of LiDAR sensors and precision of 3D model dataset.

On the contrary, learning-based approaches developed in recent years require fewer assumptions about the GNSS positioning problem, can provide solutions to mitigate dynamic urban errors, and achieve good positioning performances in complex urban environments. For example, the work in Siemuri et al. (2021) proposed to train a weighted combination of Linear Regression, Bayesian Ridge Regression, and Neural Network to predict the GNSS positioning correction. DL models are also effective for GNSS positioning correction, e.g., using Transformer (Kanhere et al. 2022) to consider attention weights between satellites and using Graph Convolutional Neural Networks (GCNN) (Mohanty and Gao 2022) to exploit topological information from constellation observations, which predict positioning correction values in complex urban environments for every time step by using reference points as supervision. However, these works simply concentrate PRR and LOS vectors as one-view GNSS inputs, ignoring relationships between different GNSS features, which are insufficient to model precise vehicle states, and these temporally continuous inputs are highly correlated, leading to disturbances in training and positioning correction. Moreover, DL models consider the localization problem of each position discretely, ignoring sequential relationships between positions.

[1] https://gssc.esa.int/navipedia/index.php/Carrier-smoothing_of_code_pseudoranges.

Additionally, DRL models are usually trained with temporal differences of values estimated from adjacent observations, helping the DRL agent to understand the surrounding environment temporally. For example, Zhang and Masoud (2020) proposed to employ a DRL algorithm, Asynchronous Advantage Actor-Critic (A3C) for the positioning correction task, which only uses historically predicted latitude and longitude to estimate belief states, and predicts horizontal positioning correction continuously. Moreover, a DRL framework was developed by Zhao et al. (2023), which utilized LSTM block to integrate the multi-input time series observations to resolve the long-term localization problem, but it also simply concatenate representations of different inputs to estimate belief states without considering relationships. In summary, the existing learning-based methods still have limits, including ineffective belief state estimation because of insufficient observations and simple state fusion, and have not considered interference from highly correlated observations.

Multiview deep reinforcement learning GNSS positioning correction with sparse representation

In this section, we first describe the details of the designed multiview RL environment for GNSS positioning correction. We then detail how to develop the DRL model for GNSS positioning correction, which can effectively process the multiview observations based on attention-weighted multiview fusion, and use sparse coding to alleviate interference from highly correlated temporally continuous observations. Finally, the MVDRL-SR model is summarized as an algorithm.

Multiview positioning correction environment

The RL environment for the GNSS positioning correction task consists of three parts, (1) comprehensive multiview observations, (2) continuous action space, and (3) effective reward setting.

Observation setting: To model the vehicle agent in the GNSS positioning correction environment, we employ different observations to represent the state considering different views. In detail, we consider employing different features derived from GNSS measurements of a GNSS device in a certain frequency, and same-frequency sequential historical vehicle positions to form the multiview observations, i.e., $\mathbf{O} = \left\{\mathbf{o}^{\text{pos}}, \mathbf{o}^{\text{los}}, \mathbf{o}^{\text{res}}\right\}$.

1. Sequential position view $\mathbf{o}^{\text{pos}}$: we first employ model-based methods to obtain initial positions $\mathbf{pos}_t^{\text{init}}$, e.g., Earth-Centered, Earth-Fixed (ECEF) solutions $\left[x_t^{\text{init}}, y_t^{\text{init}}, z_t^{\text{init}}\right]$ from SPP method WLS, KF when there are only rover GNSS measurements, which function as coarse estimations of the receiver's position to correct. Moreover, sequential historically predicted positions $\left\{\mathbf{pos}_{t-i}^{\text{pred}}\right\}_{i=1}^{k-1}$ are employed to help the RL model understand the moving status of the vehicle, which are the corrected outputs of the RL model in the past time steps. The POS view observations at time t are then,

$$\mathbf{o}_t^{\text{pos}} := \left[\mathbf{pos}_t^{\text{init}}, \mathbf{pos}_{t-1}^{\text{pred}}, \dots, \mathbf{pos}_{t-k+1}^{\text{pred}}\right] \in \mathbb{R}^{3k}, \tag{1}$$

where k is the sequence length of position view observation.

2. Line-of-sight view $\mathbf{o}^{\text{los}}$: Similar to the conventional GNSS positioning model, which uses the estimated satellite positions $\left\{\mathbf{p}_t^{\langle i \rangle}\right\}_{i=1}^{n_t}$ on the pre-determined orbits based on timing for localization, we employ the normalized Line-Of-Sight (LOS) vector **los** referring to each satellite's relative position, to help model the correction direction in the positioning correction task. Moreover, the LOS view is also related to the elevation angle, and low elevation angles may also reflect invisible satellites in complex urban environments. The LOS vector is defined as follows,

$$\mathbf{los}_t^{\langle i \rangle} = \frac{\mathbf{p}_t^{\langle i \rangle} - \mathbf{pos}_t^{\text{init}}}{\parallel \mathbf{p}_t^{\langle i \rangle} - \mathbf{pos}_t^{\text{init}} \parallel_2}, \forall t, \forall i. \tag{2}$$

Then, we can define the LOS view observations at time t as follows,

$$\mathbf{o}_t^{\text{los}} := \left[\mathbf{los}_t^{\langle 1 \rangle}, \mathbf{los}_t^{\langle 2 \rangle}, \dots, \mathbf{los}_t^{\langle n_{\max} \rangle}\right] \in \mathbb{R}^{3n_{\max}}, \tag{3}$$

where $n_{\max}$ is the maximum visible satellite number in the vehicle trajectory. Because the number of visible satellites is usually not consistent in different vehicle trajectories, the values for non-visible satellites are filled with zero vectors 0 during processing.

3. Measure residual view $\mathbf{o}^{\text{res}}$: another important feature in the conventional GNSS pseudorange positioning model is the pseudorange, referring to the distances between the rover and different satellites. Correspondingly, we employ the pseudorange residual as one view, which is the difference between excepted pseudorange and measured pseudorange, and can help estimate the correction distances in the positioning correction model. The pseudorange residual $\text{res}_t^{\langle i \rangle}$ is defined as follows,

$$\text{res}_t^{\langle i \rangle} = \rho_t^{\langle i \rangle} - \parallel \mathbf{p}_t^{\langle i \rangle} - \mathbf{pos}_t^{\text{init}} \parallel_2, \forall t, \forall i. \tag{4}$$

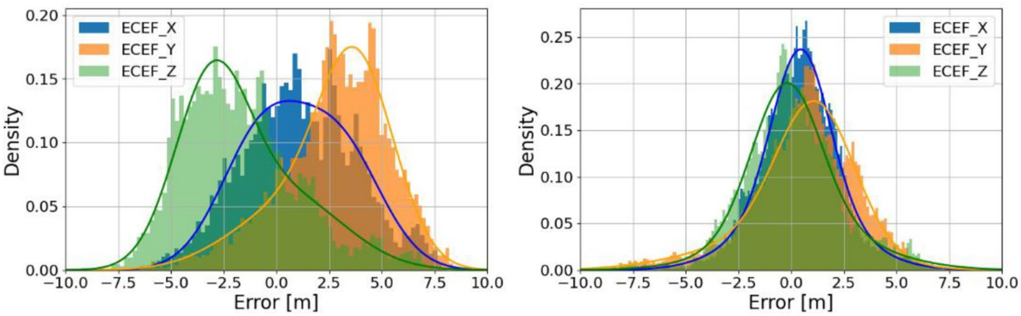


Fig. 2 Error distributions of different ECEF axes in the GSDC dataset. The left panel shows distributions in GSDC urban trajectories. The right panel shows distributions in GSDC semi-urban trajectories

Therefore, the RES view observation at time t includes all visible satellite pseudorange residuals, defined as follows,

$$\mathbf{o}_t^{\mathrm{res}} := \left[\mathrm{res}_t^{\langle 1\rangle}, \mathrm{res}_t^{\langle 2\rangle}, \ldots, \mathrm{res}_t^{\langle n_{\max}\rangle}\right] \in \mathbb{R}^{n_{\max}}. \tag{5}$$

Similarly, $n_{\max}$ is the maximum visible satellite number in the vehicle trajectory, and the values for non-visible satellites are filled with zeros during processing.

Action space setting: The action is defined as a position correction operation. To avoid complex discrete action space with many actions, which leads to learning difficulties in value estimation for each action (Lillicrap et al. 2016), we employ the continuous action setting which is sampled from the Gaussian distribution determined by the output of the actor, described as follows,

1. Define correction operations on initial positions, denoted as $\Delta\mathrm{pos}_t = \left[\Delta x_t, \Delta y_t, \Delta z_t\right]$, and thus the corrected positions from this model are,

$$\mathrm{pos}_t^{\mathrm{pred}} = \mathrm{pos}_t^{\mathrm{init}} + \Delta\mathrm{pos}_t. \tag{6}$$

2. Correction operations on each axis are, respectively, sampled from different distributions, i.e., $\Delta x_t = \mathcal{N}\left(\mu_{x_t}, \sigma^2_{x_t}\right), \Delta y \sim \mathcal{N}\left(\mu_{y_t}, \sigma^2_{y_t}\right), \Delta z_t \sim \mathcal{N}\left(\mu_{z_t}, \sigma^2_{z_t}\right)$, where all sampling is further restricted by a maximum absolute value m, i.e., $\Delta x_t = \begin{cases} \Delta x_t, & \Delta x_t < m \\ \mathrm{sign}\left(\Delta x_t\right)m, & \text{otherwise} \end{cases}$.
3. The action from the actor output is then defined as $\mathbf{a} = \left(\mu_{x_t}, \sigma_{x_t}, \mu_{y_t}, \sigma_{y_t}, \mu_{z_t}, \sigma_{z_t}\right)$, from which the RL model can obtain continuous correction operations on model-based position estimations.

One reason to set the action distribution in Gaussian distribution is that dynamic noise errors because of many different and independent factors can usually be modeled by multiple Gaussian distributions based on statistics. We further show the error distribution of different ECEF axes in the GSDC dataset used in the experiment section in Fig. 2. We can see that the overall error distributions consist of different and multiple Gaussian distributions in different trajectories, where the learning models need to learn the specific distribution outputs based on observations in different trajectories.

Reward setting: To ensure the reward can guide the correction policy learning effectively, we use a correction advantage error instead of simple positioning accuracy mean squared error (MSE) (Zhang and Masoud 2020). The correction advantage error for time t is defined as:

$$\begin{aligned} r_t := \alpha_1 \Big(&\mathsf{Haversine}\left(\left(\mathsf{Lat}_t^{\mathrm{init}}, \mathsf{Lon}_t^{\mathrm{init}}\right), \left(\mathsf{Lat}_t^{\mathrm{ref}}, \mathsf{Lon}_t^{\mathrm{ref}}\right)\right) \\ &-\mathsf{Haversine}\left(\left(\mathsf{Lat}_t^{\mathrm{pred}}, \mathsf{Lon}_t^{\mathrm{pred}}\right), \left(\mathsf{Lat}_t^{\mathrm{ref}}, \mathsf{Lon}_t^{\mathrm{ref}}\right)\right)\Big) \\ &+ \alpha_2\left(\left|\mathsf{Alt}_t^{\mathrm{init}} - \mathsf{Alt}_t^{\mathrm{ref}}\right| - \left|\mathsf{Alt}_t^{\mathrm{pred}} - \mathsf{Alt}_t^{\mathrm{ref}}\right|\right) \in \mathbb{R}, \end{aligned} \tag{7}$$

where $\mathsf{Haversine}$ is the distance formula to calculate horizontal errors. α_1 and α_2 are two parameters to adjust the effect of geodetic correction and elevation correction. Different from the MSE setting where rewards are all negative and better positioning leads rewards closer to 0, the advantage reward setting provides positive rewards when corrections are effective, making it easier for the agent to understand effective policy during learning. The Geodetic coordinates (GEO) are transformed from ECEF coordinates, i.e., $\mathbf{llh}_t = \mathcal{T}_{\mathrm{ECEF}\rightarrow\mathrm{GEO}}\left(\mathbf{pos}_t\right) = \left(\mathsf{Lat}_t, \mathsf{Lon}_t, \mathsf{Alt}_t\right)$, obtaining latitude, longitude, and altitude. Moreover, $\mathbf{llh}_t^{\mathrm{ref}} = \mathcal{T}_{\mathrm{ECEF}\rightarrow\mathrm{GEO}}\left(\mathbf{pos}_t^{\mathrm{ref}}\right)$ is the reference position for the model to learn, which can be obtained by map-matching algorithms (Quddus et al. 2007), or accurate centimeter-level positioning systems using a GNSS-INS integrated NovAtel SPAN system (Fu et al. 2020). In this reward setting, we can set different scaling parameters for horizontal and altitude errors in case of different value scales in geodetic surface and elevation errors.

Therefore, an appropriate multiview positioning correction environment is defined with corresponding comprehensive multiview observations, continuous action space, and an effective reward setting. An appropriate learned policy that can obtain high rewards in this environment can lead to good positioning correction performances in vehicle trajectories.

Multiview actor-critic learning with sparse representations

Because real-world GNSS positioning consists of complex dynamic errors, the observations cannot fully describe the agent states, we model the sequential positioning correction problem by Partially Observable Markov Decision Process (POMDP) (Hausknecht and Stone 2015; Singh et al. 2021) and employ actor-critic learning to model the complex environment errors, which can choose appropriate position correction actions by actor output for each time step. By interacting with the proposed environment with a certain policy, we can collect a sequence of observations, actions, and rewards of different time t from the proposed environment defined in the previous subsection, i.e., $\{\left(\mathbf{O}_t, \mathbf{a}_t, r_t, \mathbf{O}_{t+1}\right)\}_{t=1}^{T}$, where the sequence length is T. The target of the proposed model is to learn a policy $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{O})$, which can give the probability of choosing action $\mathbf{a}$ given observation $\mathbf{O}$, consisting of a parameter set $\boldsymbol{\theta}$, to maximize the cumulative reward which indicates the sequential correction accuracy. To this end, we need to solve mainly three problems:

1. build a belief state estimator to obtain representations from multiview sequential observations to reflect the states of the vehicle agent better.
2. develop an actor-critic network to estimate values accurately by mitigating interference from highly correlated observations based on sparse coding.
3. Optimize the parameter set $\boldsymbol{\theta}$ in the actor-critic POMDP model and output actions for continuous positioning correction.

Attention Weighted Multiview State Estimator: Considering we have a M view sequential observations $\{\mathbf{O}_t\}_{t=1}^{T} = \left\{\mathbf{o}_t^{\langle m\rangle} \in \mathbb{R}^{n^{\langle m\rangle}} : m = 1, \dots, M\right\}$, where $M = 3$ in the previous environment setting. To effectively extract features from the M view sequential observations, we employ M LSTM modules to separately process different view observations. Denoting the parameter set for m-th view as $\boldsymbol{\theta}^{\langle m\rangle}$, and outputs of m-th LSTM module as $\mathbf{h}_t^{\langle m\rangle} \in \mathbb{R}^{n_M}$, the forward propagation for m-th observation is then as follows, including block input, input gate, forget gate, cell and hidden output gate,

$$\mathbf{z}_t^{\langle m\rangle} = \tau\left(\mathbf{W}_z^{\langle m\rangle}\mathbf{o}_t^{\langle m\rangle} + \mathbf{R}_z^{\langle m\rangle}\mathbf{h}_{t-1}^{\langle m\rangle} + \mathbf{b}_z\right), \tag{8}$$

$$\mathbf{i}_t^{\langle m\rangle} = \tau\left(\mathbf{W}_i^{\langle m\rangle}\mathbf{o}_t^{\langle m\rangle} + \mathbf{R}_i^{\langle m\rangle}\mathbf{h}_{t-1}^{\langle m\rangle} + \mathbf{p}_i^{\langle m\rangle} \cdot \mathbf{c}_{t-1}^{\langle m\rangle} + \mathbf{b}_i\right), \tag{9}$$

$$\mathbf{f}_t^{\langle m\rangle} = \tau\left(\mathbf{W}_f^{\langle m\rangle}\mathbf{o}_t^{\langle m\rangle} + \mathbf{R}_f^{\langle m\rangle}\mathbf{h}_{t-1}^{\langle m\rangle} + \mathbf{p}_f^{\langle m\rangle} \cdot \mathbf{c}_{t-1}^{\langle m\rangle} + \mathbf{b}_f\right), \tag{10}$$

$$\mathbf{c}_t^{\langle m\rangle} = \mathbf{z}_t^{\langle m\rangle} \cdot \mathbf{i}_t^{\langle m\rangle} + \mathbf{c}_{t-1}^{\langle m\rangle} \cdot \mathbf{f}_t^{\langle m\rangle}, \tag{11}$$

$$\mathbf{h}_t^{\langle m\rangle} = \tanh\left(\mathbf{c}_t^{\langle m\rangle}\right) \cdot \tau\left(\mathbf{W}_h^{\langle m\rangle}\mathbf{o}_t^{\langle m\rangle} + \mathbf{R}_h^{\langle m\rangle}\mathbf{h}_{t-1}^{\langle m\rangle} + \mathbf{p}_h^{\langle m\rangle} \cdot \mathbf{c}_t^{\langle m\rangle} + \mathbf{b}_h\right), \tag{12}$$

where τ denotes the nonlinear activation, the m-th view parameter set $\boldsymbol{\theta}^{\langle m\rangle}$ includes input observation weight set $\left\{\mathbf{W}^{\langle m\rangle} \in \mathbb{R}^{n_M \times n^{\langle m\rangle}}\right\}$, recurrent state weight set $\left\{\mathbf{R}^{\langle m\rangle} \in \mathbb{R}^{n_M \times n^{\langle m\rangle}}\right\}$, peephole weight set $\left\{\mathbf{p}^{\langle m\rangle} \in \mathbb{R}^{n_M}\right\}$, and bias set $\left\{\mathbf{b}^{\langle m\rangle} \in \mathbb{R}^{n_M}\right\}$.

After obtaining representations for different view observations, we employ the attention modules to process each view and effectively fuse multiview representations with attention weights.

$$\text{att}_t^{\langle m\rangle} = \text{sigmoid}\left(\mathbf{w}^{\langle m\rangle}\mathbf{o}_t^{\langle m\rangle}\right) \in R, \tag{13}$$

where the learnable attention parameter $\mathbf{w}^{\langle m\rangle} \in \mathbb{R}^{1\times n^{\langle m\rangle}}$, which is also in the m-th view parameter set $\boldsymbol{\theta}^{\langle m\rangle}$, decides the value scale of attention weights $\text{att}_t^{\langle m\rangle}$ for m-th view representations, and the sigmoid function ensures the attention weights are in [0, 1]. The belief state is then formed by concatenating M view representations to describe better the state of the vehicle agent in the RL environment, i.e.,

$$\mathbf{h}_t = \left[\text{att}_t^{\langle 1\rangle}\mathbf{h}_t^{\langle 1\rangle}, \dots, \text{att}_t^{\langle M\rangle}\mathbf{h}_t^{\langle M\rangle}\right] \in \mathbb{R}^{Mn_M}, \tag{14}$$

where attention weights can adjust value scales of different view representations, to help the agent decide which view needs more attention during learning. Therefore, the RL policy inputs are then changed from partial observations to belief states which are supposed to be closer to fully observable environment states.

.*Actor Network and Critic Network*: After obtaining belief states $\mathbf{h}_t$, we can use the MDP-based actor-critic learning to process and learn from the belief state trajectories $\{\left(\mathbf{O}_t, \mathbf{h}_t, \mathbf{a}_t, r_t, \mathbf{O}_{t+1}\right)\}_{t=1}^{T}$. We use two DNN modules to form actor and critic, respectively, consisting of L_a and L_c layer neural networks, i.e., we have the actor parameter set $\boldsymbol{\theta}_a = \left\{\mathbf{W}_a^{(1)}, \dots, \mathbf{W}_a^{(L_a)}; \mathbf{b}_a^{(1)}, \dots, \mathbf{b}_a^{(L_a)}, \mathbf{W}_a\right\}$ and the critic parameter sets $\boldsymbol{\theta}_c = \left\{\mathbf{W}_c^{(1)}, \dots, \mathbf{W}^{(L_a)}; \mathbf{b}_c^{(1)}, \dots, \mathbf{b}_c^{(L_c)}, \mathbf{w}_c\right\}$. Denoting the l-th layer network outputs are, respectively $\mathbf{h}_a^{(l)} \in \mathbb{R}^{n_a^{(l)}}$ and $\mathbf{h}_c^{(l)} \in \mathbb{R}^{n_c^{(l)}}$, in time step t, we can obtain hidden representations of actor and critic, respectively,

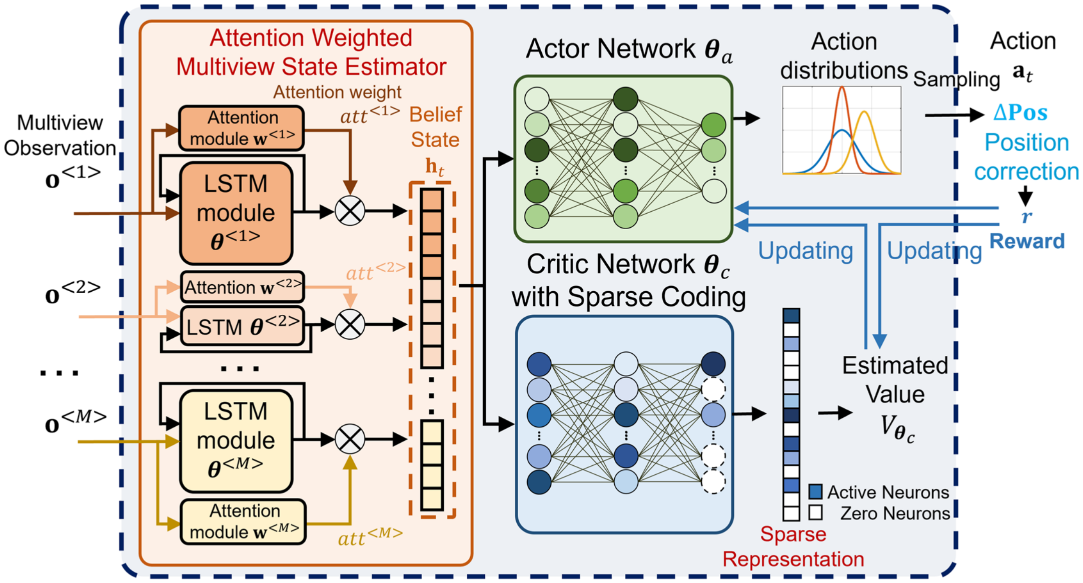


Fig. 3 Detailed architecture of the multiview actor-critic learning components in the proposed MVDRL-SR model for sequential positioning correction

$$\mathbf{h}_a^{(L_a)} = \mathbf{W}_a^{(L_a)}\tau\left(\dots\tau\left(\mathbf{W}_a^{(1)}\mathbf{h}_t + \mathbf{b}_a^{(1)}\right)\dots\right) + \mathbf{b}_a^{(L_a)} \in \mathbb{R}^{n_a(L_a)}, \tag{15}$$

$$\mathbf{h}_c^{(L_c)} = \mathbf{W}_c^{(L_c)}\tau\left(\dots\tau\left(\mathbf{W}_c^{(1)}\mathbf{h}_t + \mathbf{b}_c^{(1)}\right)\dots\right) + \mathbf{b}_c^{(L_c)} \in \mathbb{R}^{n_c(L_c)}, \tag{16}$$

where τ denotes the nonlinear activation. The output layer of the actor network needs to be the same as the action space in the environment setting, i.e., for a continuous positioning correction space defined in the previous subsection,

$$\mathbf{a}_t = \mathbf{W}_a\mathbf{h}_a^{(L_a)} = \left[\mu_{x_t},\sigma_{x_t},\mu_{y_t},\sigma_{y_t},\mu_{z_t},\sigma_{z_t}\right], \tag{17}$$

where $\mathbf{W}_a \in \mathbb{R}^{6\times n_a(L_a)}$, μ and σ are means and deviations of each dimension. For the continuous positioning correction environments, the actor network outputs action distributions for multiview observations, and decides actions from the continuous action space by sampling the distribution.

On the other hand, the critic network predicts values for different multiview observations to guide the training of the actor network. To alleviate interference in value estimation from highly correlated observations, we enforce sparsity in representations of the critic by employing the ℓ_1 norm function as the sparsity regularizer, which can induce sparsity effectively beyond the other convex functions like ℓ_2 norm (Zhao et al. 2022),

$$s\left(\mathbf{h}_c^{(L_c)}\right) = \left\|\mathbf{h}_c^{(L_c)}\right\| = \sum_j \left|(h_c^{(L_c)})_j\right|. \tag{18}$$

Denoting the output after previous network propagation as $\hat{\mathbf{h}}_c^{(L_c)}$, the sparse coding problem is then formed as,

$$\min_{\mathbf{h}_c} \hat{\mathcal{L}}(\mathbf{h}_c) = \frac{1}{2}\left\|\mathbf{h}_c - \hat{\mathbf{h}}_c^{(L_c)}\right\|_F^2 + \lambda s(\mathbf{h}_c) \tag{19}$$

By calculating the differential of $\hat{\mathcal{L}}$ respect to $\hat{\mathbf{h}}_c^{(L_c)}$, i.e.,

$$\nabla_{\mathbf{h}_c}\hat{\mathcal{L}}(\mathbf{h}_c) = \mathbf{h}_c - \hat{\mathbf{h}}_c^{(L_c)} + \lambda\,\mathrm{sign}\left(\hat{\mathbf{h}}_c^{(L_c)}\right), \tag{20}$$

Consequently, we can use the soft thresholding function to obtain the nonconvex sparse coding solutions as sparse representations of the critic,

$$\mathbf{h}_c^{(L_c)} = \mathrm{Prox}_\lambda\left(\hat{\mathbf{h}}_c^{(L_c)}\right) = \mathrm{sign}\left(\hat{\mathbf{h}}_c^{(L_c)}\right)\max\left(\left|\hat{\mathbf{h}}_c^{(L_c)}\right| - \lambda, 0\right). \tag{21}$$

Therefore, we can estimate the multiview observation values by the output sparse representations of the critic network, i.e.,

$$V^\pi(\mathbf{O}_t) := V_{\boldsymbol{\theta}}(\mathbf{O}_t) = V_{\boldsymbol{\theta}_c}(\mathbf{h}_t) = \mathbf{w}_c^\top\mathbf{h}_c^{(L_c)}, \tag{22}$$

where $\mathbf{w}_c \in \mathbb{R}^{n_c(L_c)}$, and sparse representations $\mathbf{h}_c^{(L_c)}$. can help mitigate interference and reduce coherence from correlated sequential observations.

In summary, we present the detailed architectures of the proposed multiview actor-critic learning model in Fig. 3, including the main components, i.e., multiview state estimator, actor network, and critic network.

Positioning correction algorithm

To ensure the multiview actor-critic learning functions efficiently, we employ the benchmark policy gradient method Proximal Policy Optimization (PPO) (Schulman et al. 2017) for actor-critic training, which ensures the trust region update with an advantage clipping strategy and is efficient in continuous-action environments. In the POMDP model with belief states, we employ the generalized advantage estimation (Schulman et al. 2015) to reduce variance from the noisy A^{π} is then $\hat{A}$ in time t and guide the actor training, defined as follows,

$$\hat{A}_t = \sum_{i=t}^{T-1} (\gamma \rho)^{i-t} \delta_i, \tag{23}$$

where $\delta_t = r_t + \gamma V_{\boldsymbol{\theta}_c}\left(\mathbf{h}_{t+1}\right) - V_{\boldsymbol{\theta}_c}\left(\mathbf{h}_t\right)$ is a time-difference reward residual considering the from the critic in adjacent time steps, helping the agent to consider temporally connections.

Then, the loss function for the actor network $\mathcal{L}_a$ is defined by the clipping PPO objective function as follows,

$$\mathcal{L}_a = \frac{1}{N} \sum_{i}^{N} \min\left(r_i\left(\boldsymbol{\theta}_a\right)\hat{A}_t, \mathsf{clip}\left(r_i\left(\boldsymbol{\theta}_a\right), 1 \pm \epsilon\right)\right)\hat{A}_t, \tag{24}$$

where $r_i\left(\boldsymbol{\theta}_a\right) = \pi_{\boldsymbol{\theta}_a}(\mathbf{a}_i|\mathbf{h}_i)/\pi_{\boldsymbol{\theta}_a^{\text{old}}}(\mathbf{a}_i|\mathbf{h}_i)$ is the probability ratio.

On the other hand, to achieve accurate value estimation and long-term positioning accuracy, the critic loss function $\mathcal{L}_c$ is defined by the Mean-Squared Return Error (MSRE) (Le et al. 2017), which considers the accumulated expected return reward, detailed as follows,

$$\mathcal{L}_c = \sum_{i}^{N} \frac{1}{2} \left\| g_{i+1} - w_c^{\top} h_c^{(L_c)} \right\|_2^2 \tag{25}$$

where $g_{t+1} = \sum_{i=t}^{t+T} \gamma^{i-t} r_{i+1}$ is the accumulated discount reward considering the T steps vehicle trajectory. Different from common single-step errors used in most learning-based methods, MSRE helps the agent consider correction accuracy temporally.

Furthermore, the entropy loss function $\mathcal{L}_e$ is applied to guarantee that the agent is sufficiently exploratory in its interaction with the environment,

$$\mathcal{L}_e = -\sum_{\mathbf{a}_t \in A} \pi\left(\mathbf{a}_t|\mathbf{h}_t;\boldsymbol{\theta}_a\right) \cdot \ln \pi(\mathbf{a}_t|\mathbf{h}_t;\boldsymbol{\theta}_a). \tag{26}$$

Finally, we obtain the total objective function:

$$\mathcal{L} = \mathbb{E}\left[-\mathcal{L}_a + \beta_1 \mathcal{L}_c + \beta_2 \mathcal{L}_e + \lambda s\left(\mathbf{h}_c^{(L_c)}\right)\right], \tag{27}$$

where β_1, β_2, and λ are coefficients. By interacting with the positioning correction environment, the agent can collect experiences for training the overall parameter set $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}^{\langle 1 \rangle}, \ldots, \boldsymbol{\theta}^{\langle M \rangle}, \boldsymbol{\theta}_a, \boldsymbol{\theta}_c\right\}$ with stochastic gradient descent (SGD)-based backpropagation, since all loss components are differentiable. For the nonconvex sparsity regularizer, we further employ the proximal operator in Eq. (21) to calculate thresholding sparse solutions.

In the end, the MVDRL-SR algorithm for GNSS positioning correction is formed, which can learn a stable policy, and output actions for GNSS positioning correction. the training procedure of the MVDRL-SR algorithm is summarized in Algorithm 1.

Algorithm 1 MVDRL-SR algorithm for multiview continuous positioning correction

```
Input: exploration step number $N$ , batch size $t$, discount rate $\gamma$, step number for reward accumulation $T$ , sparsity parameter $\lambda$, and learning rate $lr$.
Initialization: the multiview state estimator parameter sets $\boldsymbol{\theta}^{\langle 1\rangle},\ldots,\boldsymbol{\theta}^{\langle M\rangle}$ , the critic parameter set $\boldsymbol{\theta}_c$, the actor parameter set $\boldsymbol{\theta}_a$, and initial epoch number $e=0$.
Main iterations:
Repeat
    $e=e+1$;
    Collect trajectories $\{(\mathbf{O}_i,\mathbf{h}_i,\mathbf{a}_i,r_i,\mathbf{O}_{i+1})\}_{i=1}^{N}$ with the multiview state estimator and the actor policy with parameters $\boldsymbol{\theta}^{\langle 1\rangle},\ldots,\boldsymbol{\theta}^{\langle M\rangle}$ and $\boldsymbol{\theta}_a$ from the environment;
    Compute the accumulated discount rewards for collected trajectories,
        $g_{t+1}=\sum_{i=t}^{t+T}\gamma^{i-t}r_{i+1}$
    Compute observation values $V_{\boldsymbol{\theta}_c}(\mathbf{h}_t)$ with critic parameters $\boldsymbol{\theta}_c$ and proximal operator in Eq. 26;
    Compute the generalized advantage estimation $\hat{A}_t$ based on temporal differences of estimated observation values;
    Draw $t$ tuples $\{(\mathbf{O}_i,\mathbf{h}_i,\mathbf{a}_i,r_i,\mathbf{O}_{i+1})\}_{i=1}^{t}$ from the dataset randomly;
    Update all parameters $\boldsymbol{\theta}^{\langle 1\rangle},\ldots,\boldsymbol{\theta}^{\langle M\rangle}$, $\boldsymbol{\theta}_a$, and $\boldsymbol{\theta}_c$ based on an SGD method by calculating gradients from the total loss function $\mathcal{L}$ in Eq. 32;
Until stopping criterion is met.
Output: $\boldsymbol{\theta}=\{\boldsymbol{\theta}^{\langle 1\rangle},\ldots,\boldsymbol{\theta}^{\langle M\rangle},\boldsymbol{\theta}_a,\boldsymbol{\theta}_c\}$
```

Experimental validation

In this section, we validate the proposed approach using two real-world GNSS datasets. Firstly, we detail the validation setup for the two datasets. Secondly, we introduce compared model-based and state-of-the-art learning-based methods and then analyze the performances in terms of positioning error. All experiments in this section were performed via PyTorch 1.8, and run on a CPU with 2.6 GHz AMD cores and 256G RAM. All average positioning performances in the experiment section use Root Mean Square Errors (RMSE) as the metric.

Dataset and initialization

One of the two real-world GNSS datasets is formed from the Android Raw GNSS Measurements Dataset which was used in the Google Smartphone Decimeter Challenge (GSDC) 2022 (Fu et al. 2020). Another one is formed by our collected rover and base GNSS dataset in Guangzhou, China (GZGNSS). For both two GNSS datasets, we randomly select half of the trajectories for training DRL algorithms, and present correction tests comparison on the left unseen trajectories in different times and routes after training. The training and testing separation are the same for all compared algorithms. Moreover, satellite positions $\left\{\mathbf{p}_t^{\langle i\rangle}\right\}_{i=1}^{n_t}$ are all estimated by GNSS broadcast measurements. All compared algorithms are implemented by the authors through turning corresponding parameters and employing the same ECEF correction labels as the proposed model to obtain their optimal performances for comparison.

*GSDC Dataset***:** The GSDC dataset consists of GNSS measurements collected by different Android smartphones from various driving trajectories in the San Francisco Bay Area and Los Angeles, in which the high-accuracy positions $\mathrm{pos}_t^{\mathrm{ref}}$ are obtained by a centimeter-level GNSS-INS integrated NovAtel SPAN system (Fu et al. 2020). Because the number of visible satellites continuously changes during each trajectory, we exclude trajectories with zero visible satellites in certain time steps, and use the left 79 trajectories with full reference positions as the dataset. We separate these trajectories into two settings for validation, i.e., (1) the GSDC urban trajectory dataset includes 32 trajectories with many buildings next.to roads or overpasses during the trajectory, resulting in relatively higher distance errors, and (2) the GSDC semi-urban trajectory dataset includes 47 trajectories with few buildings by the road and fewer distance errors. An urban area example trajectory near Los Angeles and a semi-urban example trajectory near Stanford are, respectively, shown in the left and right panels in Fig. 4.

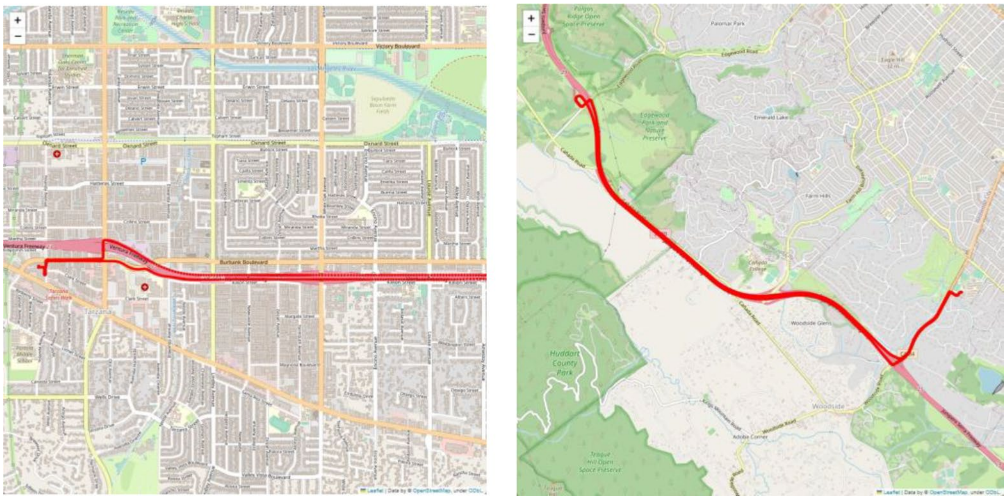


Fig. 4 Vehicle trajectory examples in the GSDC 2022 dataset. The left panel shows an urban area example in the northwest of Los Angeles. The right panel shows a semi-urban example near Stanford

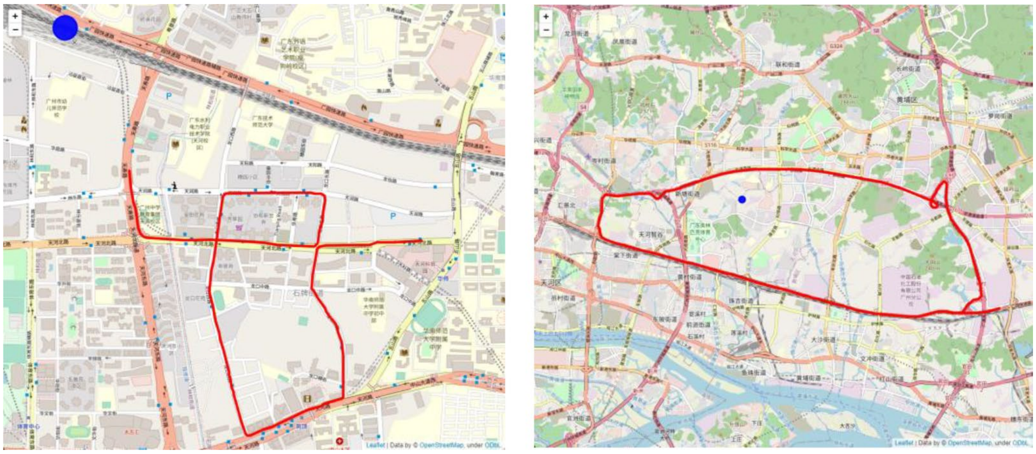


Fig. 5 Vehicle trajectory examples in the GZGNSS dataset. The red line indicates the rover positioning trajectory on the moving vehicle, and the blue circle is the location of the base receiver. The left panel shows an urban area example in the central areas of Guangzhou. The right panel shows a semi-urban example on the highway of Guangzhou

As the GSDC dataset only has the smartphone rover GNSS measurements, we employ the model-based SPP method WLS+KF to obtain the initial solutions $\mathrm{p}_t^{\mathrm{init}}$ at different times t, which function as the baseline for positioning correction. For GNSS features, we only choose L1C frequency signals of GPS to form LOS vectors and pseudorange residuals in consideration of simplifying processing since it is most consistent during all trajectories. Since then, we have $n_{\max} = 32$ for both different views.

*GZGNSS Dataset***:** The GNSS measure.ents are collected by multiple N307-5D GNSS receivers in different trajectories in Guangzhou areas, where one static receiver is regarded as the base and another dynamic rover receiver is placed on the moving vehicle. Similarly, the high-accuracy. positions $\mathbf{pos}_t^{\mathrm{ref}}$, which are regarded as the ground truth positions, are obtained by a centimeter-level GNSS-INS integrated NovAtel SPAN system. Moreover, we also separate collected trajectories into two settings for validation, i.e., (1) the GZGNSS urban trajectory dataset including 54 trajectories with 400–2600 time steps, and (2) the GZGNSS semi-urban trajectory dataset including 62 trajectories with 300–3400 time steps. An urban example in the central areas of Guangzhou and a semi-urban example on the semi-urban highway of Guangzhou are, respectively, shown in the left and right panels in Fig. 5.

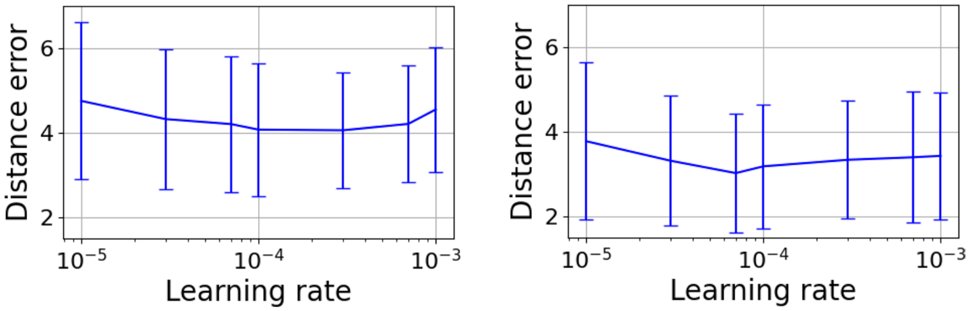


Fig. 6 Performances of the proposed MVDRL-SR algorithm with a range of learning rates in different positioning correction environments. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

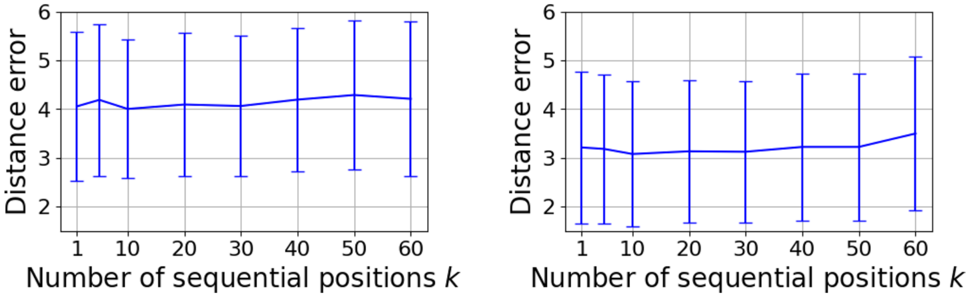


Fig. 7 Performances of the proposed MVDRL algorithm with different k in sequential position observations in different positioning correction environments. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

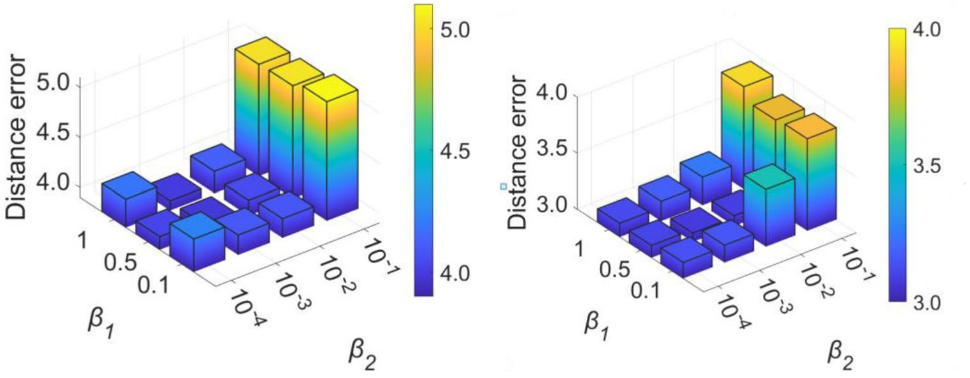


Fig. 8 Performances of the proposed MVDRL-SR algorithm with a range of β_1. and β_2. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

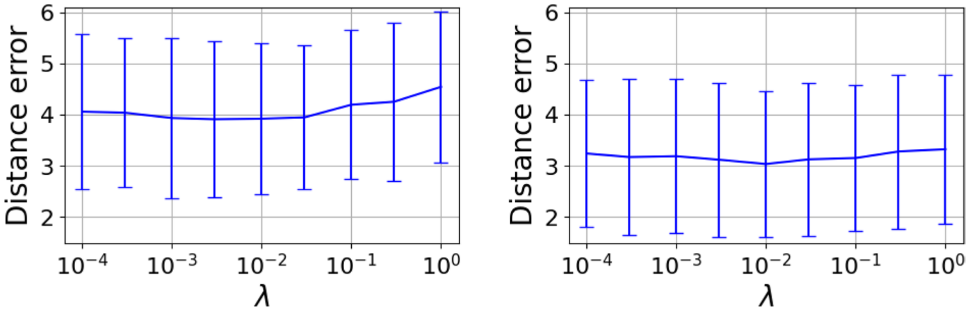


Fig. 9 Performances of the proposed MVDRL-SR algorithm with a range of λ in different positioning correction environments. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

Since we have both base and rover GNSS measurements in our collected GZGNSS dataset, we employ the carrier-phase differential Real-Time Kinematic positioning (RTK) solutions to obtain initial solutions p_t^{init} as the baseline of the proposed MVDRL-SR model. The measure residual view observations $\mathbf{o}^{\text{res}}$ use carrier-phase residuals in this dataset. Moreover, we select two frequencies of three constellations to form the GNSS observation in our GZGNSS dataset, i.e., BDS-L2I, BDS-L5Q, GPS-L1C, GPS-L5Q, GAL-L1C, and GAL-L5Q.

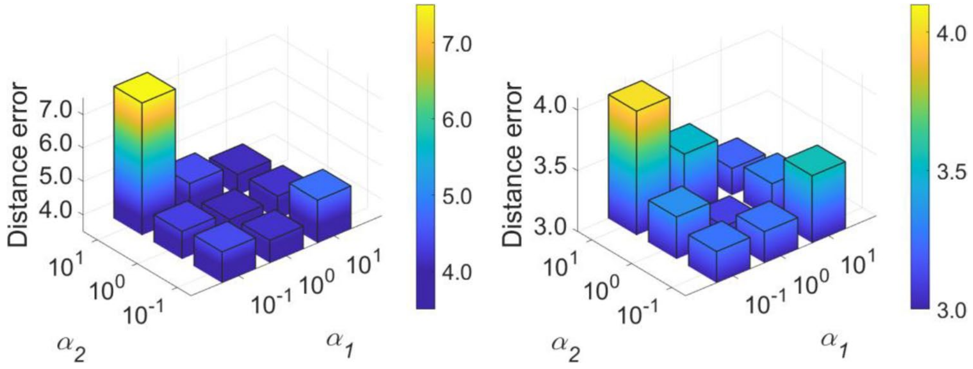


Fig. 10 Performances of the proposed MVDRL-SR algorithm with a range of α_1 and α_2. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

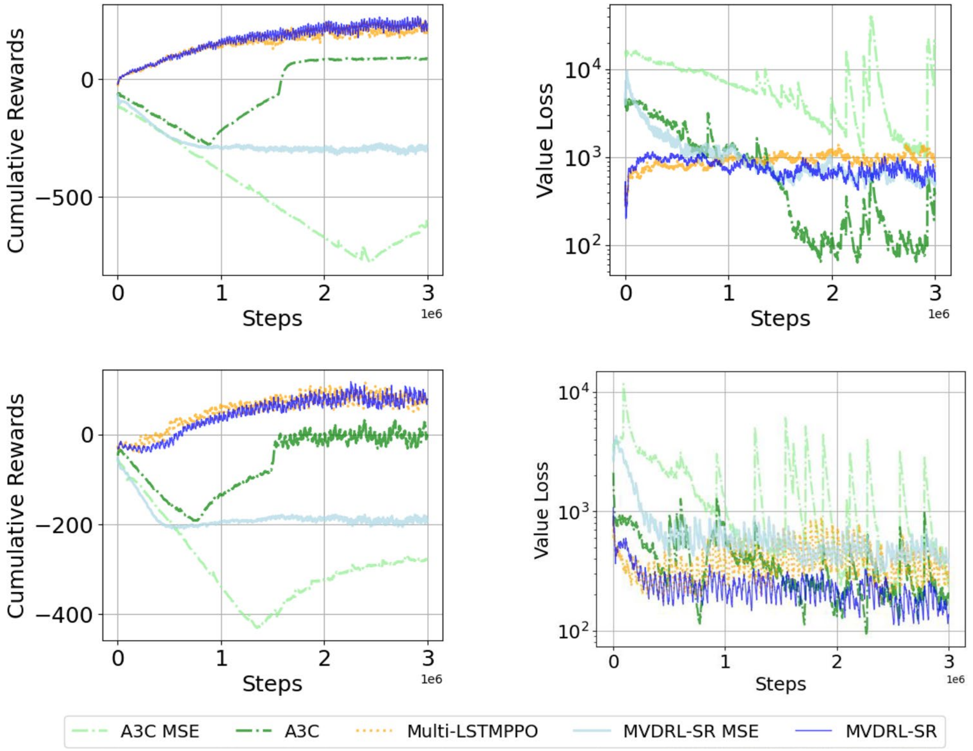


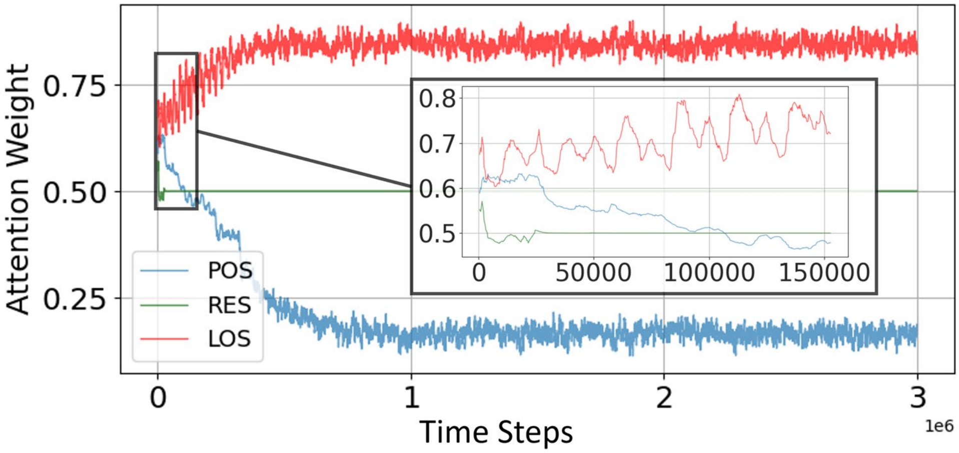
Fig. 11 Convergence curves of control and training performances for different DRL algorithms with corresponding optimal parameters in the GSDC trajectory dataset. The top two panels show performances in GSDC urban trajectories. The bottom two panels show performances in GSDC semi-urban trajectories

Parameter selection

In this subsection, we validate how parameters affect the performances of the proposed MVDRL-SR. We first present the performances of the proposed model by tuning the learning rates in Fig. 6. Overall, the proposed MVDRL-SR can obtain optimal localization performances with learning rates around 10^{-4}. In detail, the optimal learning rates in GSDC semi-urban are slightly smaller than those in GSDC urban.

One tunable parameter in the multiview environment is the number of sequential positions k, which affects the sequence length and ratio of POS observation in the observations, we then present how it affects the localization performances in Fig. 7. Overall, we can see that different k may affect the localization performances in different environments. Moreover, an appropriate k value around 10 can help the MVDRL model, which is a trimmed version of the proposed MVDRL-SR without the sparse coding part, to obtain better final positioning correction performances. Furthermore, we can also see that too large k. may affect the overall performances in different datasets.

In the loss function Eq. (27), we first tune the β_1 for critic loss and β_2 for entropy loss of the proposed model, shown in Fig. 8. Based on the grid search results, we select $\beta_1 = 0.5$ and $\beta_2 = 10^{-3}$ in the following experiments. Then, λ, which affects the sparsity of representations in the critic, is crucial to the performance of the proposed model. As presented in Fig. 9 with a range of λ, overall, an appropriate λ around 10^{-2} can help enhance the localization performances in both GSDC urban and semi-urban. However, larger λ than 10^{-1} may cause some reduction in



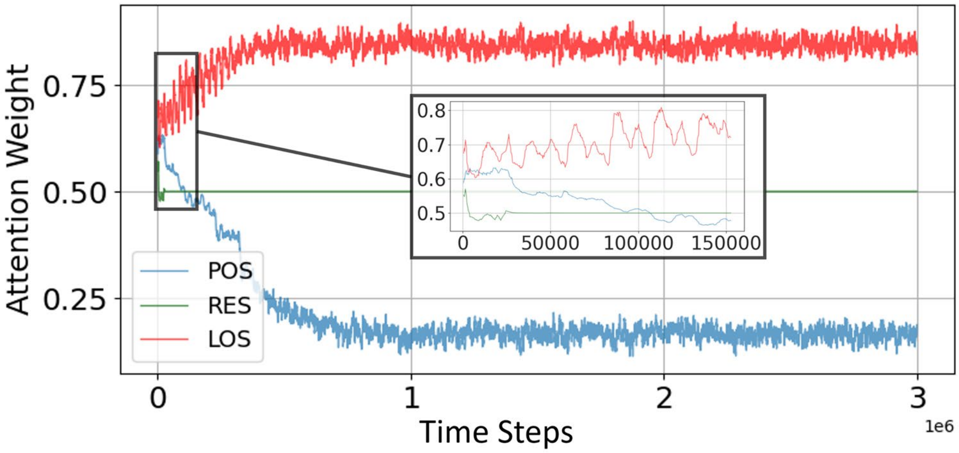


Fig. 12 Learned attention weights of different views of the proposed MVDRL-SR in the GSDC trajectory dataset. The top panel shows performances in GSDC urban trajectories. The bottom panel shows performances in GSDC semi-urban trajectories

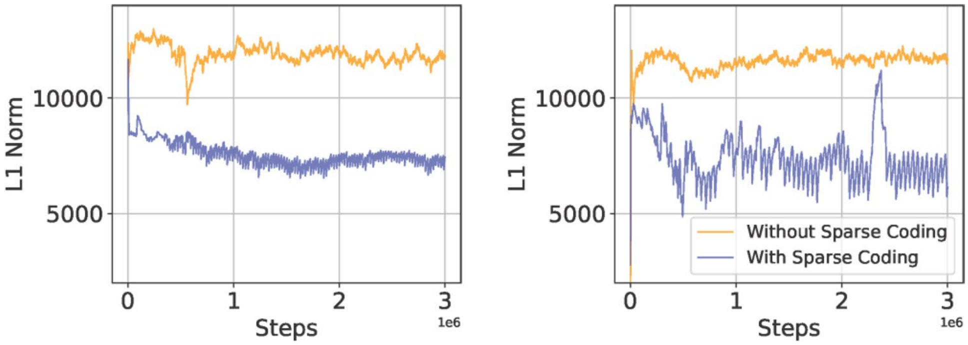


Fig. 13 Representation sparsity in terms of the ℓ_1 norm of the proposed MVDRL-SR with or without sparse coding in the GSDC trajectory dataset. The left panel shows performances in GSDC urban trajectories. The right panel shows performances in GSDC semi-urban trajectories

localization performances, which can be because too high sparsity causes information losses in representations.

Furthermore, we have grid-searched the parameter for the proposed reward setting in Eq. (7), i.e., α_1 for horizontal advantage errors and α_2 for altitude advantage errors, shown in Fig. 10. Overall, when selecting $\alpha_1 = 1$ and $\alpha_2 = 1$ in the GSDC datasets, the proposed MVDRL-SR can obtain optimal correction performances in both urban and semi-urban trajectories. One reason for this is the horizontal and altitude error distributions are similar in the GSDC dataset, with a horizontal/altitude error ratio from 0.71 to 1.33, and thus an equal parameter setting can help the agent learn an effective correction policy.

Performance comparison in the GSDC dataset

After validating optimal parameters for MVDRL-SR in the previous subsection, we compare the performances with state-of-the-art positioning correction algorithms in the GSDC dataset. We first present the convergence performance of the proposed algorithm MVDRL-SR and the other DRL-based positioning correction methods in Fig. 11.

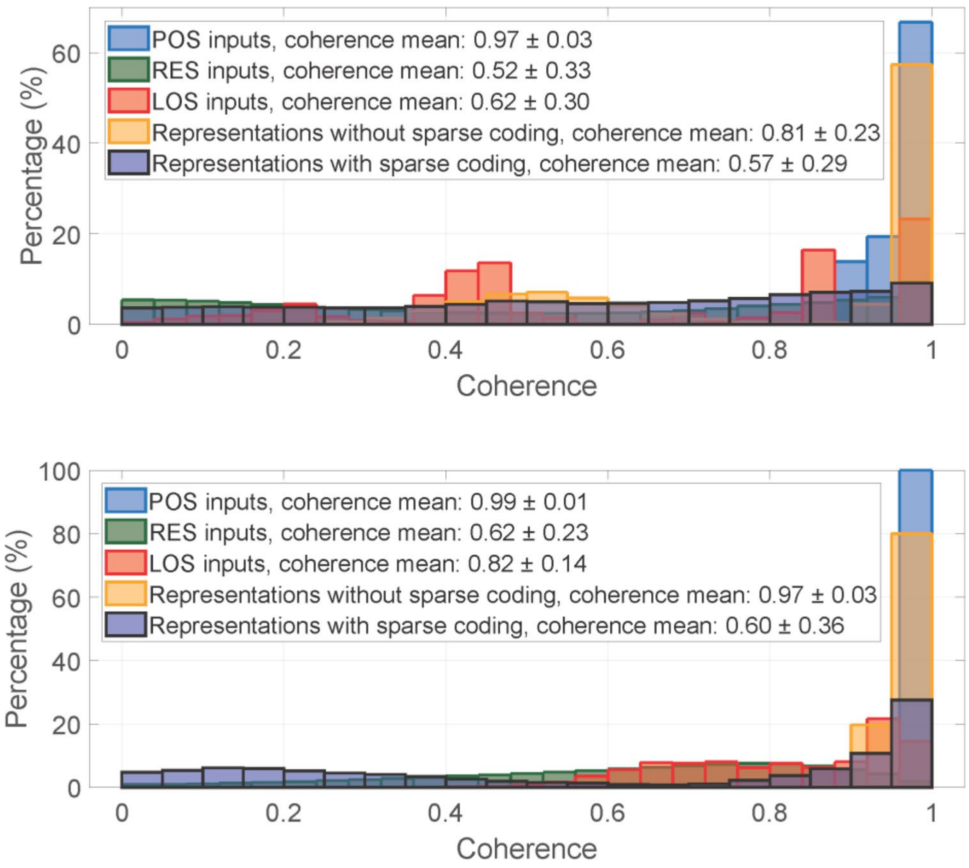


Fig. 14 Coherence distribution of the proposed MVDRL-SR with or without sparse coding in the GSDC trajectory dataset. The top panel shows performances in GSDC urban trajectories. The bottom panel shows performances in GSDC semi-urban trajectories

Table 1 Average testing positioning performances (meters) of different methods with corresponding optimal parameters in GSDC urban and semi-urban trajectory datasets

Method	X_{ECEF}	Y_{ECEF}	Z_{ECEF}	Distance error	Horizontal error	Altitude error
(a) *GSDC urban trajectories*						
WLS+KF	2.33 ± 1.25	3.23 ± 1.57	2.79 ± 1.38	5.39 ± 1.43	3.06 ± 1.00	3.81 ± 1.49
A3C MSE	2.38 ± 1.41	2.24 ± 1.51	2.81 ± 1.37	4.89 ± 1.46	3.66 ± 1.21	2.75 ± 1.51
A3C	2.33 ± 1.24	2.21 ± 1.37	2.81 ± 1.37	4.77 ± 1.38	3.06 ± 1.01	3.10 ± 1.49
SetTransformer	2.24 ± 1.39	2.07 ± 1.39	2.33 ± 1.43	4.41 ± 1.47	2.98 ± 0.97	2.76 ± 1.57
GCNN	2.10 ± 1.41	2.02 ± 1.38	2.24 ± 1.45	4.20 ± 1.57	2.85 ± 1.10	2.61 ± 1.62
Multi-LSTMPPO	2.06 ± 1.26	2.04 ± 1.29	2.28 ± 1.42	4.18 ± 1.50	2.77 ± 1.00	2.69 ± 1.54
MVDRL	1.91 ± 1.16	2.11 ± 1.35	2.08 ± 1.27	4.00 ± 1.42	2.67 ± 0.98	2.56 ± 1.48
MVDRL-SR MSE	2.02 ± 1.25	2.05 ± 1.34	2.17 ± 1.41	4.12 ± 1.49	2.72 ± 1.01	2.59 ± 1.52
MVDRL-SR	**1.94 ± 1.06**	**2.02 ± 1.28**	**2.05 ± 1.25**	**3.94 ± 1.30**	**2.64 ± 0.93**	**2.48 ± 1.45**
Method	X_{ECEF}	Y_{ECEF}	Z_{ECEF}	Distance error	Horizontal error	Altitude error
(b) *GZGNSS semi-urban trajectories*						
WLS+KF	1.53 ± 1.07	2.12 ± 1.34	1.80 ± 1.14	3.57 ± 1.65	1.95 ± 1.03	2.69 ± 1.65
A3C MSE	1.70 ± 1.05	2.01 ± 1.35	1.97 ± 1.19	3.70 ± 1.65	2.00 ± 1.02	2.83 ± 1.67
A3C	1.45 ± 1.04	2.12 ± 1.40	1.78 ± 1.11	3.52 ± 1.66	1.92 ± 1.01	2.66 ± 1.63
SetTransformer	1.48 ± 1.09	2.02 ± 1.39	1.80 ± 1.14	3.48 ± 1.71	1.88 ± 1.03	2.64 ± 1.68
GCNN	1.46 ± 1.11	1.99 ± 1.43	1.81 ± 1.20	3.44 ± 1.77	1.81 ± 1.06	2.68 ± 1.75
Multi-LSTMPPO	1.36 ± 1.01	1.88 ± 1.36	1.80 ± 1.15	3.14 ± 1.58	1.79 ± 0.97	2.32 ± 1.61
MVDRL	1.36 ± 0.98	1.80 ± 1.19	1.67 ± 1.07	3.08 ± 1.49	1.77 ± 1.05	2.27 ± 1.50
MVDRL-SR MSE	1.36 ± 0.98	1.76 ± 1.21	1.64 ± 1.06	3.12 ± 1.49	1.76 ± 0.92	2.31 ± 1.51
MVDRL-SR	**1.36 ± 0.95**	**1.72 ± 1.17**	**1.62 ± 1.05**	**3.01 ± 1.40**	**1.72 ± 0.88**	**2.21 ± 1.44**

Overall, all algorithms can converge in reasonable numbers of time steps within 3×10^6 in both GSDC urban and semi-urban. The training convergence curves of all three algorithms are not smooth with some oscillation, suggesting that the distributions of trajectories are varied from each other. In detail, A3C can converge faster than the other two algorithms, but the cumulative rewards, which refer to control and localization performances, are much lower. Moreover, although the reward convergence curves of MVDRL-SR are similar to those of Multi-LSTMPPO in both GSDC urban and semi-urban datasets, MVDRL-SR can obtain smaller value losses than Multi-LSTMPPO in both cases, suggesting that sparse representations can help MVDRL-SR to predict state values more accurately. Furthermore, we have shown two algorithms with the MSE reward setting (Zhang and Masoud 2020) as ablation tests. Overall, the two algorithms with the advantage error setting can converge faster with lower value losses than corresponding algorithms with the MSE reward setting, suggesting that the critic can learn the state value distribution faster and better with the advantage error reward setting.

In Fig. 12, we then illustrate how the learned attention weights are distributed, which can help the agent understand relationships between different views, and fuse the multiview information more effectively. As the x-axis is the time step, the attention module processes observations of different time steps and learns relatively stable scaling attentions for different views during training. In general, the attention module can effectively analyze relationships between different views and output attention weights, where the attention weights of POS-view representations can generally converge after 10^6 time step-training, while weights of the LOS view need about half of 10^6 steps, and the RES view needs smaller than 50,000 steps. Moreover, the learned attention weights of LOS-view and RES-view representations are much larger than those of POS-view representations which are constantly below 0.2, suggesting that LOS-view and RES-view representations are considered more informative than POS-view by the attention module. Furthermore, after the fast convergence of the RES view, the attention weight value is then constantly at 0.5 for RES representations, two possible reasons are: (1) the information of RES observations in different trajectories is considered similar, (2) there returns little gradient in neurons relating to RES observations, suggesting they are not very helpful in positioning correction.

We then show how sparse representations are learned during training, in terms of ℓ_1 norm sparsity of representations in Fig. 9, and the coherence distribution of representations in Fig. 10. The ℓ_1 norm, which is the sum of absolute values of representations, is both an intuitive sparsity measurement and also the regularization in the proposed MVDRL-SR model. Overall, the ℓ_1 norm values of MVDRL-SR with sparse coding can generally converge at 7000 within 10^6. time steps, and are much smaller than those without sparse coding, suggesting the effectiveness of obtaining sparse representations using the proposed model in both GSDC urban and semi-urban trajectories. Benefiting from the sparsity in representations, one effect is lower coherence distributions shown in Fig. 10. Besides the RES view, the other two views of temporally and spatially continuous observations, i.e., vehicle positions POS and relatively satellite positions LOS, are highly correlated in both GSDC urban and semi-urban datasets. Moreover, the learned representations of MVDRL-SR can effectively reduce coherence in both two GSDC datasets, while representations without sparse coding are still highly correlated (Figs. 13 and 14).

In Table 1, we summarize the testing performances of different positioning algorithms in GSDC urban and semi-urban datasets. Overall, the proposed MVDRL-SR can obtain better testing localization performances in terms of average RMSE in different coordinates in both two datasets than the others, including model-based WLS+KF; Deep learning-based SetTransformer, GCNN; RL-based A3C, Multi-LSTMPPO. In detail, compared with the model-based SPP baseline, MVDRL-SR can improve about 27% in GSDC urban trajectories, and about 16% in GSDC semi-urban trajectories. Moreover, the two RL-based methods, MVDRL-SR and Multi-LSTMPOP can outperform the two deep learning-based methods, one reason is that the two RL-based methods consider relationships of observations in temporal. Meanwhile, we can see that the positioning correction performances in GSDC urban trajectories are generally better than those in GSDC semi-urban trajectories, which corresponds to the convergence reward values of the two datasets in Fig. 11, indicating training in GSDC semi-urban trajectories is more difficult.

For the ablation test, we present the results of the trimmed version of MVDRL-SR, i.e., MVDRL as a comparison in Table 1. Considering the two main components of the proposed methods, the improvements of MVDRL from Multi-LSTMPPO, which simply concentrates observations of different views, can support the effects of attention-weighted multiview fusion. Moreover, MVDRL-SR can obtain smaller average and deviation values of distance errors than MVDRL in both GSDC urban and semi-urban trajectories, supporting the effectiveness of the sparse coding-based critic network. Finally, benefiting from attention-weighted multi-view fusion and sparse representations, MVDRL-SR can improve about 6% positioning accuracy from Multi-LSTMPPO in the GSDC urban dataset, and about 4% in the GSDC semi-urban dataset. Moreover, we present the two algorithms with the MSE reward setting as ablation tests. Generally,

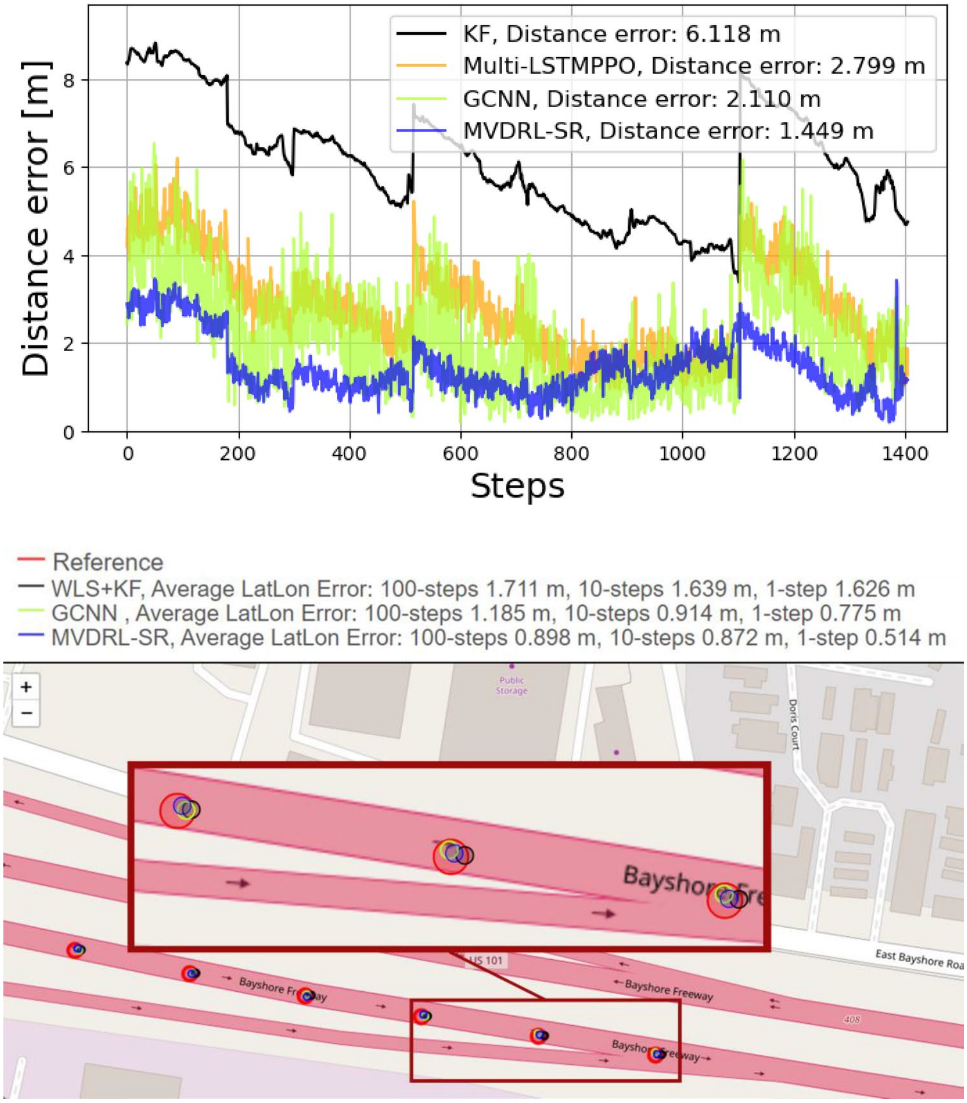


Fig. 15 Vehicle positioning correction performances of different algorithms in a GSDC urban trajectory. The top panel shows the overall performance throughout the trajectory. The bottom panel shows the detailed performances on the map

the two algorithms with the MSE setting obtain higher localization errors than corresponding algorithms with the advantage error setting, supporting the effectiveness of the proposed reward setting.

Intuitive comparisons of the positioning performances of different algorithms are shown in Figs. 15 and 16, about positioning accuracy in two example trajectories in the GSDC dataset. Overall, the proposed MVDRL-SR can outperform the other algorithms in most time steps, achieving smaller distance errors in the two GSDC urban and semi-urban trajectories. Moreover, the distance error oscillation of MVDRL-SR is also smaller than the other two algorithms, suggesting better stability of the proposed model. In the bottom panel of Figs. 15 and 16, we present how different algorithms perform positioning corrections on horizontal maps. Overall, the blue circles referring to the proposed MVDRL-SR are closer to the center of the red circles which are the reference points obtained by the NovAtel SPAN system (Figs. 15 and 16).

Performance comparison in the GZGNSS dataset

In this subsection, we present and analyze the performances of different algorithms in our collected GZGNSS datasets. Different from the GSDC dataset, we employ the carrier phase differential RTK solutions as the baseline to validate the proposed MVDRL-SR model.

In Table 2, we detail the average performances of different algorithms in GZGNSS urban and semi-urban trajectories. Overall, the proposed MVDRL-SR can effectively improve the localization performances and outperform the other algorithms in terms of distance errors, horizontal errors, and altitude errors in the two GZGNSS datasets. Different from the SPP method WLS+KF, the altitude errors in the baseline RTK are much higher than the

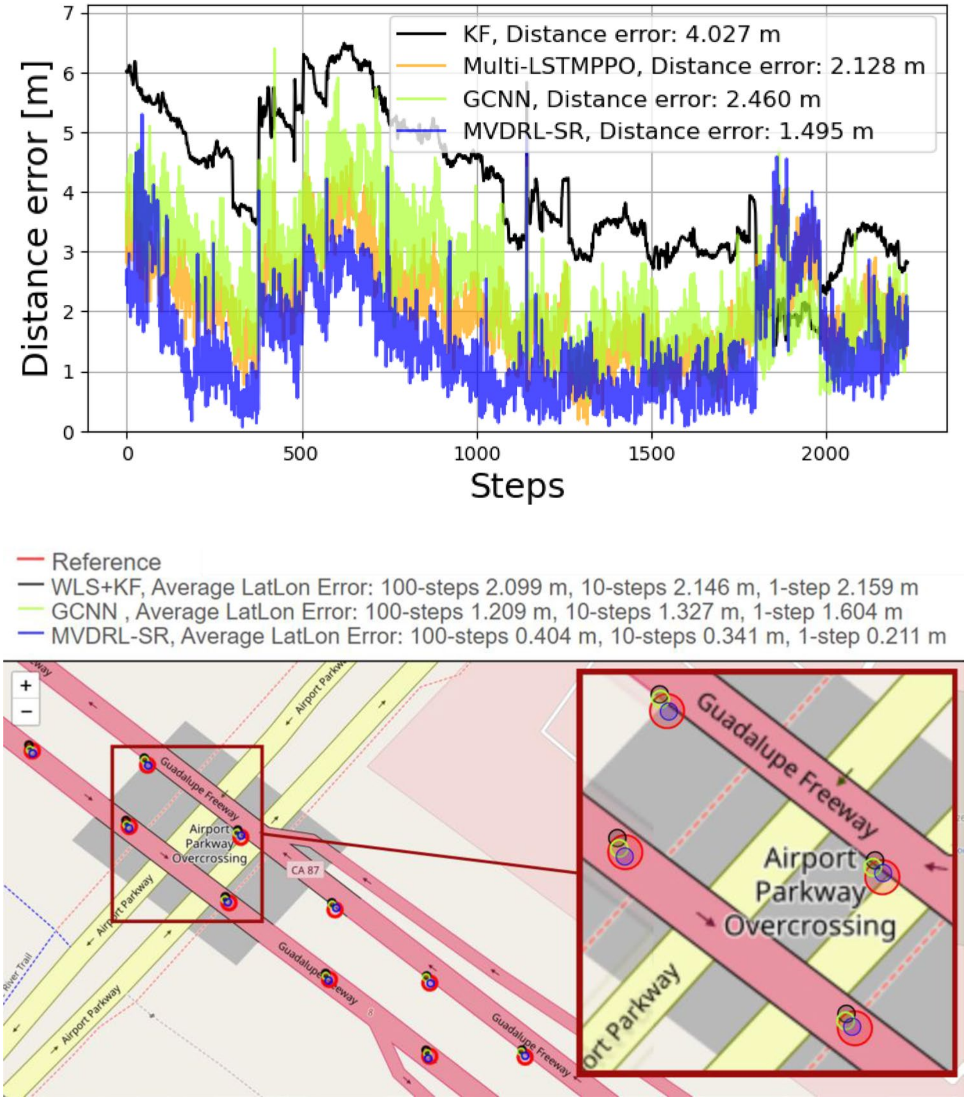


Fig. 16 Vehicle positioning correction performances of different algorithms in a GSDC semi-urban trajectory. The top panel shows the overall performance throughout the trajectory. The bottom panel shows the detailed performances on the map

horizontal errors, so we employ the reward function in Eq. (7) by setting $\alpha_1 = 1$ and $\alpha_2 = 0.1$ for all learning-based methods. In detail, MVDRL-SR can obtain about 16% better horizontal accuracy than the RTK baseline in GZGNSS urban trajectories, and about 13% better horizontal accuracy in GZGNSS semi-urban trajectories, supporting the effectiveness of the proposed model. Moreover, we can see that the horizontal accuracy of A3C is the same as the RTK baseline in GZGNSS semi-urban trajectories, which can be also seen in Fig. 17 and is mainly related to its complex discrete action space setting. Furthermore, the large altitude error in the baseline RTK can also be effectively mitigated by the proposed MVDRL-SR model, with about 45% improvement in GZGNSS urban trajectories and 77% improvement in GZGNSS semiurban trajectories (Fig. 18).

We then present the intuitive performance comparison in GZGNSS urban and semi-urban example trajectories in Figs. 17 and 18. Overall, the proposed MVDRL-SR can obtain lower horizontal errors and altitude errors than the RTK baseline, RL-based A3C, and DL-based GCNN in most time steps. Although the RTK solutions may have even higher altitude errors than WLS+KF, MVDRL-SR can effectively correct the error and obtain better average performance. Moreover, the example semi-urban trajectory is with a smaller number of high buildings, leading to fewer multipath effects and more steady RTK solutions than the urban one. Nevertheless, the average horizontal improvements from RTK in GZGNSS semi-urban trajectories are smaller than those in GZGNSS urban. One reason can be that oscillations and high errors occupy a small proportion in semi-urban trajectories, and these sudden changes in surrounding environments have not been well learned. Furthermore, the proposed MVDRL-SR can effectively correct the errors to reach closer to the centers of the red reference circles, supporting the good generalization ability of the proposed model (Fig. 17).

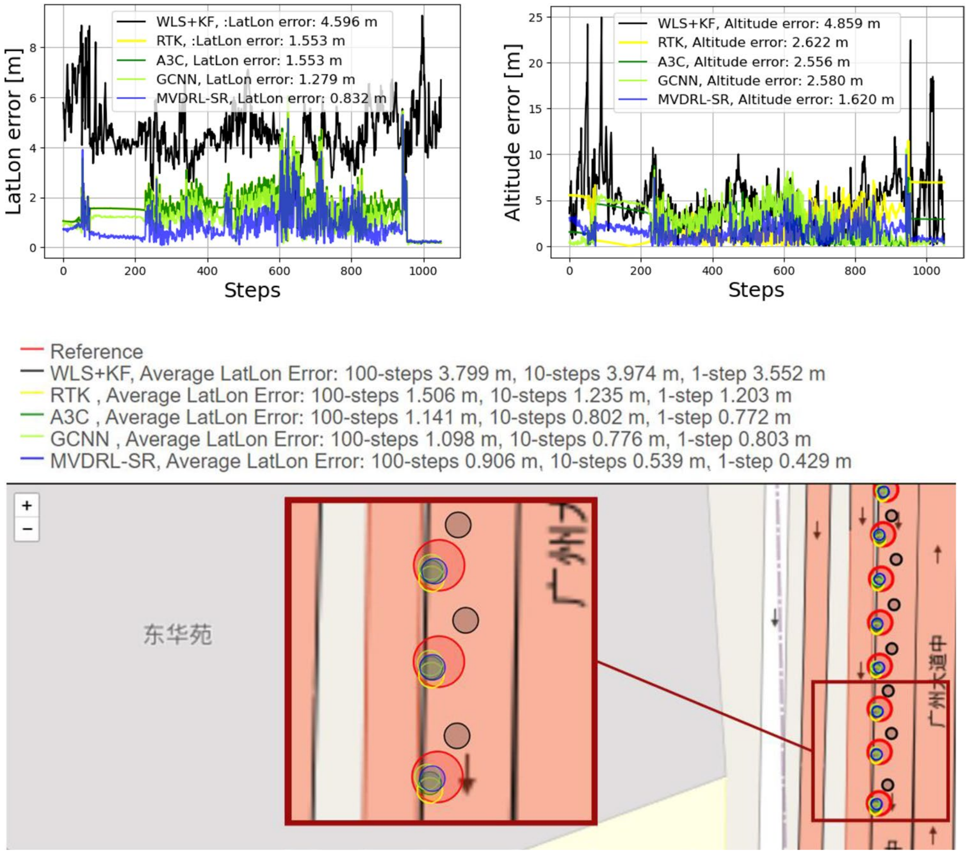


Fig. 17 Vehicle positioning correction performances of different algorithms in a GZGNSS semi-urban trajectory. The top two panels show the overall performance throughout the trajectory. The bottom panel shows the detailed performances on the map

Discussion

In the experimental validation, we have presented the performances of the proposed MVDRL-SR in different real-world GNSS datasets, and compared it with state-of-the-art approaches for GNSS positioning correction in complex urban environments. Moreover, we have also validated the effectiveness of the two main modules in MVDRL-SR, i.e., the attention-weighted multiview fusion and the sparse coding-based critic network.

1. For the attention-weighted multiview fusion, we first tuned the sequence length k of the POS view in Fig. 7, and an appropriate sequence length of 10 can achieve better positioning performance in both GSDC urban and GSDC semi-urban trajectories. For observations of different views, the attention weights can be learned adaptively and effectively, as shown in Fig. 12, all weights can converge in 10^6 time steps and help the agent to understand which view is more informative and needs more attention. Table 1 shows that the average performances of MVDRL can outperform Multi-LSTMPPO, which only concentrates multiview observations, supporting the effectiveness of the attention-weighted multiview fusion.
2. For the sparse coding-based critic network, we first tuned the regularizer parameter λ in Fig. 9, in which an appropriate λ around 10^{-2} can obtain lower distance errors in both GSDC urban and semi-urban trajectories. We then present the representation sparsity during training in Fig. 13, showing that the imposed $\ell 1$ norm can effectively reduce the sparsity of representations during training. Moreover, Fig. 14 shows that sparse coding can also effectively reduce the coherence of representations from highly correlated multiview observations. The effectiveness of the sparse coding-based critic module is also validated in Table 1 by comparing MVDRL-SR to MVDRL.
3. For the GSDC dataset, we first present the convergence of environment returns and value losses in Fig. 11, where MVDRL-SR can obtain similar converged rewards but smaller value losses than Multi-LSTMPPO. We then present performance comparisons with state-of-the-art approaches in terms of three ECEF axes errors and distance errors, shown in Table 1. The proposed MVDRL-SR can outperform the model-based baseline WLS+KF with 27% improvement in GSDC urban and 16% improvement in GSDC semi-urban. Moreover, MVDRL-SR can obtain about 6% improvement from DL-based GCNN and DRL-based Multi-LSTMPPO. In the GZGNSS dataset, we employ the carrier-differential

Table 2 Average positioning performances (meters) of different methods with corresponding optimal parameters in GZGNSS urban and semi-urban trajectory datasets

Method	X_{ECEF}	Y_{ECEF}	Z_{ECEF}	Distance error	Horizontal error	Altitude error
(a) *GZGNSS urban trajectories*						
WLS+KF	4.34 ± 2.96	5.76 ± 4.83	5.54 ± 2.87	9.99 ± 5.38	5.19 ± 2.12	7.66 ± 5.91
RTK	2.41 ± 1.77	4.92 ± 3.45	3.18 ± 1.96	6.68 ± 3.89	2.14 ± 1.96	5.98 ± 3.85
A3C	1.69 ± 1.83	3.37 ± 3.40	1.78 ± 1.90	4.56 ± 4.00	2.01 ± 1.84	3.77 ± 3.82
GCNN	1.64 ± 1.80	3.06 ± 3.52	1.73 ± 2.00	4.28 ± 4.11	2.05 ± 1.97	3.40 ± 3.91
MVDRL-SR	**1.48 ± 1.70**	**2.94 ± 3.42**	**1.60 ± 1.88**	**4.01 ± 3.99**	**1.79 ± 1.80**	**3.27 ± 3.80**
Method	X_{ECEF}	Y_{ECEF}	Z_{ECEF}	Distance error	Horizontal error	Altitude error
(b) *GZGNSS semi-urban trajectories*						
WLS+KF	2.65 ± 1.70	4.20 ± 2.49	3.79 ± 1.56	7.14 ± 2.56	4.55 ± 1.21	4.71 ± 3.06
RTK	2.37 ± 0.81	5.11 ± 1.41	2.61 ± 0.73	6.33 ± 1.55	0.89 ± 0.77	6.13 ± 1.59
A3C	1.33 ± 0.81	2.56 ± 1.38	1.23 ± 0.70	3.29 ± 1.57	0.89 ± 0.77	3.05 ± 1.53
GCNN	0.75 ± 0.83	1.33 ± 1.49	0.64 ± 0.75	1.81 ± 1.76	0.81 ± 0.80	1.49 ± 1.66
MVDRL-SR	**0.74 ± 0.73**	**1.23 ± 1.38**	**0.67 ± 0.67**	**1.75 ± 1.59**	**0.77 ± 0.70**	**1.43 ± 1.53**

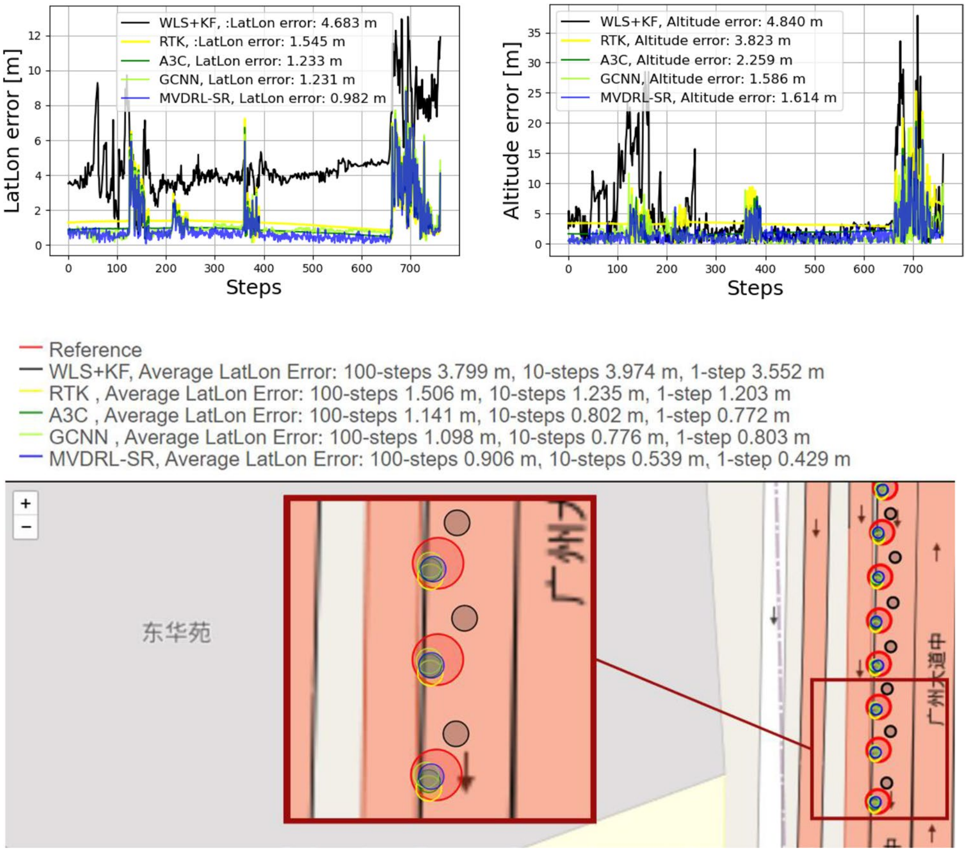


Fig. 18 Vehicle positioning correction performances of different algorithms in a GZGNSS urban trajectory. The top two panels show the overall performance throughout the trajectory. The bottom panel shows the detailed performances on the map

RTK solutions as the initial position baseline. RTK can reach centimeter-level accuracy in open environments, however, the trajectories are collected in complex urban areas in Guangzhou with a highly active ionosphere, leading to meter-level performances which are affected by ionosphere effects and inaccurate double-differenced ambiguity fixing. Again, MVDRL-SR can obtain better localization performances than the other approaches, with 16% horizontal and 45% altitude improvement from the baseline RTK, 13% horizontal improvement from the DL-based GCNN. Furthermore, we present intuitive trajectory examples of the two datasets, respectively, in Figs. 15, 16, 17 and 18.

Table 3 Average running time costs of different methods in the GSDC datasets

Method	WLS+KF	A3C	SetTransformer	GCNN	MVDRL-SR
Time (10^{-3} Sec)	9.87 ± 1.27	4.62 ± 0.25	3.04 ± 0.13	1.01 ± 0.06	5.25 ± 0.14

Overall, the proposed method MVDRL-SR can employ different model-based methods as initial position baseline, e.g., WLS, KF, and RTK, to learn a correction policy for better localization performances than model-based baseline, existing DL-based, and RL-based methods in real-world GNSS datasets in different areas. However, we should note that these learning methods are based on GNSS measurements and model-based initial positions, which means they cannot function well when there is no visible satellite. Moreover, when the data distribution in training data are varied too large, the convergence can slow down, and the model may only learn the distribution of the main parts in the dataset, where we need to learn specific models for different datasets and distributions. Furthermore, to validate the effectiveness of the proposed model in positioning after training, we show the running time costs of different methods for each time step in Table 3, noting that time costs of learning-based methods are counted when they receive model-based inputs, so the overall time costs of learning-based methods used in practice should add the time cost of employed model-based baseline. Overall, after training, all learning-based methods can output correction results within reasonable time costs from 10^{-3} to 5×10^{-3} seconds, making them possible to achieve a 50 Hz positioning.

Conclusion and future work

In conclusion, we develop a DRL framework for GNSS positioning correction in complex urban environments, i.e., MVDRL-SR, which uses attention-weighted multiview fusion to process multiview observations to represent the vehicle states sufficiently and employ the ℓ_1 norm sparse coding to promote sparse hidden representations in critic to alleviate interference from highly correlated temporally continuous observations. To effectively model the vehicle state, we construct the positioning correction RL environment with multiview observations, and develop an attention-weighted multiview fusion framework to exploit historical features from different views separately based on LSTM modules, and fuse belief states based on adaptively learned attention weights considering relationships between views. To alleviate interference from redundant and correlated multiview temporally continuous observations, we employ the ℓ_1 norm regularizer and corresponding proximal operator in the critic to promote sparse hidden representation during network propagation and effectively reduce coherence in representations to enhance the precision of value estimation. Finally, we construct the learning model based on sparse representation-driven actor-critic DRL structure and multiview fusion estimated POMDP belief states, with the cumulative value estimation and temporal difference advantage setting to help the agent consider positioning correction in temporal. In the experimental validation, we perform results in the open GSDC dataset, and our collected GZGNSS dataset, both datasets are separated as urban and semi-urban trajectories. Overall, the proposed MVDRL-SR can outperform both model-based baseline and state-of-the-art learning-based methods in different real-world GNSS datasets. For example, MVDRL-SR can obtain about 27% positioning accuracy improvement from the WLS+KF baseline in GSDC urban trajectories, about 16% horizontal accuracy improvement, and 45% altitude accuracy improvement from RTK in GZGNSS urban trajectories. Moreover, MVDRL-SR can obtain 6% lower distance errors in GSDC urban trajectories, and 13% lower horizontal errors in GZGNSS urban trajectories than DL-based GCNN. Furthermore, the ablation tests support the effectiveness of the two main modules, i.e., the attention-weighted multiview fusion can effectively learn attention weights for different views and help reduce about 4% distance errors. Besides, the sparse coding-based critic network can effectively reduce the sparsity of representations during training, reduce about 30% coherence of representations, and further improve localization performances. In future works, we will consider adding more GNSS features and inertial measurements to form more comprehensive observations in the RL environment to improve the vehicle state estimation accuracy and improve positioning correction performances.

Acknowledgements This research is supported in part by National Natural Science Foundation of China under Grants 62203122, 62273106, 62320106008, 62373114, in part by Guang Dong Basic and Applied Basic Research Foundation 2023A1515011159, 2023A1515011480, in part by China Postdoctoral Science Foundation funded project 2022M720840, and in part by National Key Research and Development Plan-Strategic Scientific and Technological Innovation Cooperation Key Project under Grant 2023YFE0209400.

Author contributions Conceptualization: Haoli Zhao, Zhenni Li; Methodology: Haoli Zhao, Zhenni Li; Formal analysis and Investigation: Haoli Zhao; Software: Haoli Zhao, Qianming Wang; Data Curation: Haoli Zhao, Qianming Wang; Writing original draft preparation: Haoli Zhao; Writing review and editing: Haoli Zhao, Zhenni Li, Kan Xie, Ci Chen; Funding acquisition: Haoli Zhao, Zhenni Li, Shengli Xie, Ming Liu; Supervision: Zhenni Li, Shengli Xie, Ming Liu.

Data Availability The data that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

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

References

Cai H, Hu Z, Huang G, Zhu D, Su X (2018) Integration of GPS, monocular vision, and high definition (HD) map for accurate vehicle localization. Sensors 18(10):3270

Fu GM, Khider M, van Diggelen F (2020) Android raw GNSS measurement datasets for precise positioning. In: Proceedings of ION GNSS 2020, Institute of Navigation, September 21–25, 1925–1937.

Groves PD, Adjrad M (2017) Likelihood-based GNSS positioning using LOS/NLOS predictions from 3d mapping and pseudoranges. GPS Solutions 21(4):1805–1816

Hausknecht M, Stone P (2015) Deep recurrent Q-learning for partially observable MDPs. In: Proceedings of the 2015 AAAI fall symposium series, pp 29–37

Kanhere AV, Gupta S, Shetty A, Gao G (2022). Improving GNSS positioning using neural-network-based corrections. Navig J Inst Navig 69(4)

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

Le L, Kumaraswamy R, White M (2017) Learning sparse representations in reinforcement learning with sparse coding. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp 2067–2073

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations, May 2–4

Liu X, Wen W, Hsu LT (2022) 3D LiDAR aided GNSS real-time kinematic positioning via coarse-to-fine batch optimization for high accuracy mapping in dense urban canyons. In: Proceedngs Of ION GNSS 2022, Institute of Navigation, September 19–23, 1954–1965

Medina D, Li H, Vila-Valls J, Closas P (2019) On robust statistics for GNSS single point positioning. In: Proceedings of 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, October, 281–3287

Mohanty A, Gao G (2022) Learning GNSS positioning corrections for smartphones using graph convolution neural networks. In: Proceedings of ION GNSS 2022, Institute of Navigation, September 19–23, 2215–2225

Quddus MA, Ochieng WY, Noland RB (2007) Current map-matching algorithms for transport applications: State-of-the art and future research directions. Transp Res Part C Emerging Technol 15(5):312–328

Schulman J, Moritz P, Levine S, Jordan, M, Abbeel, P (2015) High-dimensional continuous control using generalized advantage estimation. In: Proceedings of International Conference on Learning Representations, May 2–4

Schulman J, Wolski F, Dhariwal P, Radford, A, Klimov, O (2017) Proximal policy optimization algorithms. arXiv preprint arXiv:170706347

Shu B, Liu H, Zhang J, Pan G, Jiang J (2017) Performance assessment of partial ambiguity resolution based on BDS/GPS combined positioning. Geomatics Inf Sci Wuhan Univ 42(7):989–994

Siemuri A, Selvan K, Kuusniemi H, et al (2021) Improving precision GNSS positioning and navigation accuracy on smartphones using machine learning. In: Proceedings of ION GNSS 2022, Institute of Navigation, September 20–24, pp 3081–3093

Singh G, Peri S, Kim J, Kim, H, Ahn, S (2021) Structured world belief for reinforcement learning in pomdp. In: Proceedings of International Conference on Machine Learning, PMLR, pp 9744–9755

Skog I, Handel P (2009) In-car positioning and navigation technologies—a survey. IEEE Trans Intell Transp Syst 10(1):4–21

Teunissen P, Khodabandeh A (2015) Review and principles of PPP-RTK methods. J Geodesy 89(3):217–240

Verhagen S, Teunissen PJ (2017) Least-squares estimation and Kalman filtering. In: Springer handbook of global navigation satellite systems, pp 639–660

Wang S, Bao Z, Culpepper JS, Cong G (2021) A survey on trajectory data management, analytics, and learning. ACM Computing Surveys (CSUR) 54(2):1–36

Wen W, Zhou Y, Zhang G, Fahandezh-Saadi S, Bai XW, Zhan W, Tomizuka M, Hsu LT (2020) UrbanLoco: a full sensor suite dataset for mapping and localization in urban scenes. In: Proceedins of 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp 2310–2316

Xin S, Geng J, Zhang G, Ng HF, Guo J, Hsu LT (2022) 3d-mapping-aided PPP-RTK aiming at deep urban canyons. J Geodesy 96(10):1–19

Zhang E, Masoud N (2020) Increasing GPS localization accuracy with reinforcement learning. IEEE Trans Intell Transp Syst 22(5):2615–2626

Zhao H, Li Z, Chen W, Zheng Z, Xie S (2022) Accelerated partially shared dictionary learning with differentiable scale-invariant sparsity for multi-view clustering. IEEE Trans Neural Netw Learn Syst 34(11):8825–8839

Zhao H, Li Z, Chen C, Wang L, Xie K, Xie S (2023) Fusing vehicle trajectories and GNSS measurements to improve GNSS positioning correction based on actor-critic learning. In: Proceedings of ION ITM 2023, Institute of Navigation, January 24–26, pp 82–94

Zhu N, Marais J, Bétaille D, Berbineau M (2018) GNSS position integrity in urban environments: a review of literature. IEEE Trans Intell Transp Syst 19(9):2762–2778

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Haoli Zhao is a post-doctoral fellow in the School of Automation, Guangdong University of Technology, and in Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing(GDUT), and also with the School of Artificial Intelligence, Heibei University of Technology, China. His current research interests focus on sparse representation, reinforcement Learning, deep Learning, and GNSS positioning.

Zhenni Li is currently an associate professor in the School of Automation, Guangdong University of Technology, and in Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing(GDUT), Guangzhou 510006, China. Her research interests include machine learning, deep reinforcement learning, and GNSS positioning.

Qianming Wang is currently an engineer in the Techtotop Microelectronics Technology Co. Ltd and pursuing a Ph.D. degree in the School of Automation, Guangdong University of Technology. His current research interests include GNSS data processing, multi-source data fusion processing, machine learning, etc.

Kan Xie is currently an Associate professor in School of Automation, Guangdong University of Technology, and in 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, Guangzhou 510006, China. His research interests include machine learning, non-negative signal processing, blind signal processing, smart grid, and the Internet of Things.

Shengli Xie IEEE Fellow and Foreign Full Member (Academician) of the Russian Academy of Engineering. He is the head of Key Laboratory of Intelligent Detection and the Internet of Things in Manufacturing(GDUT) and Key Laboratory of Intelligent Information Processing and System Integration of IoT (GDUT), Ministry of Education. He has co-authored two books and over 150 research papers in refereed journals and conference proceedings and was awarded Highly Cited Researcher in 2020. His research interests include blind signal processing, machine learning, and Internet of Things. He was awarded the Second Prize of the National Natural Science Award of China in 2009.

Ming Liu is currently an assistant professor in Thrust of Robotics and Autonomous Systems, Hong Kong University of Science and Technology, Guangzhou, China. His research interest includes dynamic environment modeling, 3D mapping, machine learning, and visual control.

Ci Chen is now a professor with School of Automation, Guangdong University of Technology, and in 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, Guangzhou, China. His research interests include reinforcement learning, resilient control, and computational intelligence.