Journal of Global Positioning Systems
Vol. 22, Issue 2:196-205
DOI:10.5081/jgps.22.2.196

Mamba Augmented Adaptive Kalman Filter for Smartphone GNSS Navigation

Divya Bhavsar, Hongzhou Yang
Department of Geomatics Engineering, University of Calgary, Calgary, Alberta, Canada
✉ the corresponding author, divya.bhavsar@ucalgary.ca

***Abstract*—**Smartphone GNSS measurements exhibit time-varying errors caused by multipath, non-line-of-sight reception, changing satellite availability, and device-dependent measurement characteristics. This study investigates a Mamba-augmented adaptive Kalman filter for group-level measurement covariance scaling in post-processed smartphone GNSS positioning. Using a centered 14-epoch window containing 20 measurement and contextual features, the Mamba selective state-space model predicts six variance scale factors for the GPS, GLONASS, and Galileo code and carrier-phase observation groups. These factors modify the variances produced by the existing C/N_0 -weighted RTKLIB stochastic model, while the process noise covariance and remaining positioning configuration are unchanged. The model is trained using ground-truth-referenced residuals from the Google Smartphone Decimeter Challenge (GSDC) 2022 training data, whereas test inference uses only features generated by the standard RTKLIB processing pipeline. The final configuration achieved a GSDC 2022 private-leaderboard score of 1.533 m, which would retrospectively correspond to third position in the published competition standings. Under a matched experimental configuration, Transformer and LSTM backbones obtained scores of 1.608 m and 1.626 m, respectively. Mamba also reduced trajectory-level RMS horizontal error for 59 of the 109 evaluated training drive-phone trajectories, although the benefit varied across trajectories and error metrics. The result was achieved using a GNSS-only differential post-processing framework without IMU integration, sensor fusion, or additional hardware. These findings demonstrate the potential of selective state-space modelling as a data-driven approach to temporal covariance adaptation in post-processed smartphone GNSS positioning.

***Keywords*—**Smartphone GNSS, Adaptive covariance scaling, Mamba selective-state-space model, Adaptive Kalman Filtering, Mamba-AKF framework, RTKLIB, Post-processed kinematic positioning

1. INTRODUCTION

Global Navigation Satellite Systems (GNSS) support a wide range of positioning and navigation applications, including intelligent transportation and location-based services. The availability of raw GNSS measurements from Android devices since Android 7.0 has enabled smartphones to become accessible platforms for advanced positioning research (Banville & Van Diggelen, 2016; Zangenehnejad & Gao, 2021). Nevertheless, high-accuracy smartphone GNSS positioning remains challenging. Compared with geodetic-grade receivers, smartphones commonly exhibit higher code noise, weaker and more variable signal strength, degraded carrier-phase quality, and device-dependent measurement irregularities (Paziewski et al., 2019; Shinghal & Bisnath, 2021; Zangenehnejad & Gao, 2021). These effects become particularly important in urban and suburban environments, where signal blockage, non-line-of-sight (NLOS) reception, and multipath propagation can produce rapidly changing measurement errors (Groves et al., 2013; Zhou et al., 2023).

GNSS positioning performance depends strongly on the stochastic model used to represent pseudorange and carrier-phase measurement uncertainty. Conventional models commonly determine measurement variances using satellite elevation angle, carrier-to-noise-density ratio (C/N□), or a combination of both (Li et al., 2022). These approaches provide physically interpretable and computationally efficient weighting strategies and therefore remain important reference models for GNSS processing. However, the relationship between measurement quality, satellite geometry, receiver motion, and the surrounding environment can be complex. Studies have consequently explored site-specific and environment-dependent weighting strategies to represent errors that are not described completely by a single elevation- or C/N_0 -based relationship (Zhang et al., 2018; Zhang et al., 2022).

Within Kalman-filter-based positioning, the measurement noise covariance matrix (R) determines the relative confidence assigned to the observations, whereas the process noise covariance matrix (Q) represents uncertainty in the system model. Adaptive estimation methods update these covariance quantities as measurement conditions change. Innovation-based adaptive estimation (IAE), for example, estimates covariance information from innovation sequences (Yin et al., 2021), while least-squares variance component estimation (LS-VCE) provides a statistically rigorous framework for estimating observation variances

(Teunissen & Amiri-Simkooei, 2008). LS-VCE has also been applied specifically to stochastic modelling of smartphone GNSS observations (Zangenehnejad & Gao, 2023). These methods differ in their statistical assumptions and data requirements, and their performance can depend on factors such as observation redundancy, estimation-window design, and the quality of the available innovation or residual sequences. The present study does not seek to replace these methods; instead, it investigates a complementary data-driven strategy for learning temporal covariance adjustments from measurement and contextual features.

Machine-learning methods provide another way to represent relationships between GNSS measurements and changing propagation conditions. Previous studies have applied support vector machines and neural networks to multipath and LOS/NLOS classification using raw-measurement indicators, pseudorange consistency features, or signal-correlator outputs (Hsu, 2017; Suzuki & Amano, 2021). Deep learning has also been combined with statistical testing for smartphone GNSS outlier detection (Jiang et al., 2025). Data-augmentation approaches have also been investigated to improve classification performance under imbalanced LOS/NLOS datasets (Nie & Yang, 2025). Classification and outlier-detection methods are valuable when severely degraded measurements should be identified or excluded. Continuous covariance scaling offers a related but different strategy: rather than making only a discrete acceptance or rejection decision, it adjusts the influence of an observation group within the stochastic model. The two strategies should therefore be regarded as complementary rather than competing formulations.

GNSS measurement conditions also exhibit temporal structure. Signal quality and residual behavior may persist across neighboring epochs as the receiver moves through areas with trees, buildings, overpasses, or other obstructions. Sequential models can use this local context instead of treating each epoch independently. Recurrent neural networks, including long short-term memory (LSTM) models, process such temporal dependencies through recurrent state updates. Transformer architectures use self-attention to model relationships across a sequence and have also been investigated for GNSS signal classification (Li et al., 2023). More recently, the Mamba selective state-space model has provided an alternative sequence-modelling architecture with linear complexity in sequence length (Gu & Dao, 2023). These architectural properties motivate an investigation of Mamba for GNSS temporal modelling, but they do not by themselves establish that Mamba is more accurate or faster than recurrent or attention-based models for the relatively short sequences considered in this work. Accordingly, controlled LSTM and Transformer comparisons are included in the experimental evaluation.

This study investigates a Mamba-augmented adaptive Kalman filter for post-processed smartphone GNSS positioning. The model uses a centered temporal window of smartphone-derived measurement and contextual features to estimate six covariance scale factors corresponding to GPS, GLONASS, and Galileo code and carrier-phase observation groups. These factors are applied on top of the existing C/N_0-weighted RTKLIB stochastic model, thereby adding learned temporal adaptation without replacing the underlying positioning framework. The model is trained using residuals generated with reference to the ground-truth positions available for the GSDC 2022 training data, while inference on the test trajectories uses only features produced by the standard RTKLIB processing pipeline. Model configuration, including feature selection, loss weights, and temporal-window length, is determined empirically using the GSDC benchmark objective.

The principal contribution of this work is therefore an empirical investigation of selective state-space modelling for group-level adaptive covariance scaling in a post-processed smartphone GNSS pipeline. The evaluation considers positioning performance, learned feature attribution, temporal alpha behavior, architecture comparisons with LSTM and Transformer backbones, and computational characteristics on the experimental workstation. The results establish performance relative to the implemented C/N_0-weighted RTKLIB baseline and the evaluated neural architectures; they are not intended to establish general superiority over all statistical adaptive-filtering or stochastic-modelling methods.

Section 2 describes the feature-generation pipeline, covariance-scaling formulation, temporal model, loss function, and integration with RTKLIB. Section 3 presents the experimental setup, ablation studies, architecture and efficiency comparisons, positioning results, and interpretation analyses. Section 4 summarizes the conclusions, limitations, and directions for future work.

2. Methodology

This study investigates a Mamba-based temporal model for adaptive measurement covariance scaling in post-processed smartphone GNSS positioning. From a centered sequence of measurement and contextual features, the model predicts six variance scale factors for the GPS, GLONASS, and Galileo code and carrier-phase observation groups. These factors multiply the variances produced by the existing C/N□-weighted RTKLIB stochastic model, while the process noise covariance matrix Q and the remaining positioning configuration are unchanged.

The framework is evaluated using the GSDC 2022 dataset. Ground-truth-referenced residuals are used to construct the training data, whereas test features are generated exclusively from the standard RTKLIB processing outputs. The following subsections describe feature extraction, temporal-window construction, the Mamba architecture, the training objective, and integration with RTKLIB.

2.1 Dataset and Feature Extraction

The GSDC 2022 dataset contains 62 training drives with ground-truth positions and 36 test drives for which evaluation is performed through the Kaggle leaderboard (Howard et al., 2022). The trajectories cover open-sky, suburban, and urban environments in California and contain raw GNSS measurements collected using multiple smartphone models.

The trajectories are initially processed in post-processed kinematic mode using the C/N□-weighted RTKLIB baseline adapted from Tim Everett's open-source GSDC solution. This baseline includes RTKLIB's signal-strength-dependent variance model and measurement-screening settings but does not include the learned scale factors. Observations from nearby continuously operating reference stations are used for differential processing.

RTKLIB solution and measurement outputs are parsed to obtain receiver velocity, satellite count, state-uncertainty quantities, C/N_0, and code and carrier-phase residuals. Horizontal velocity is calculated from the eastward and northward velocity components. The measurement variables are aggregated by constellation to form GPS, GLONASS, and Galileo feature channels.

For the training set, residuals are evaluated with reference to the available ground-truth positions. Ground truth is used only for constructing training residuals and evaluating training trajectories. For the test set, residual and contextual features are obtained from the standard RTKLIB solution because test ground truth is unavailable.

The final representation contains 14 numerical features and six one-hot-encoded device indicators, resulting in 20 features per epoch. The input features and preprocessing methods are listed in Table 1. StandardScaler is applied to velocity, satellite count, and state-uncertainty variables, while RobustScaler is applied to C/N_0 and residual features. All scaling parameters are fitted exclusively on the training dataset and applied unchanged to the test dataset. Consequently, no test-set statistics or ground-truth information are used during test preprocessing.

2.2 Temporal Window Construction

Each drive-phone trajectory is sorted chronologically and divided into overlapping 14-epoch windows with a stride of one epoch. A prediction is assigned to the eighth epoch of each window, providing seven preceding epochs and six following epochs around the assigned epoch. The resulting centered-window formulation is non-causal and is intended for post-processed positioning rather than real-time filtering.

The 14-epoch length was selected empirically from the evaluated window sizes using the GSDC benchmark score. The residual autocorrelation analysis provides additional evidence that the residual sequences contain temporal dependence over neighboring epochs, although it is not used to claim that 14 is a uniquely optimal statistical length.

Each model input has the shape $(B, 14, 20)$, where B is the batch size. Residual channels and satellite counts are retained separately for loss evaluation. Windows remain associated with their original drive-phone trajectories so that temporal smoothness is evaluated only between adjacent windows from the same trajectory.

Table 1. Input Features for Mamba Model

Feature Type	Features	Scaling Method
Kinematics	$Velocity_mps$	$StandardScalar$
Satellite Count	Total_Sat_Count	$StandardScalar$
Uncertainty	$sdn, sde, sdne$	$StandardScalar$
Signal Strength	$CN0_GPS$, $CN0_GLO$, $CN0_GAL$	$RobustScalar$
Code Residuals	$Code_Res_GPS$, $Code_Res_GLO$, $Code_Res_GAL$	$RobustScalar$
Phase Residuals	$Phase_Res_GPS$, $Phase_Res_GLO$, $Phase_Res_GAL$	$RobustScalar$
Device Identity	$GooglePixel4$, $GooglePixel4XL$, $GooglePixel5$, $GooglePixel6Pro$, $SamsungGalaxy S20Ultra$, $XiaomiMi8$	$None$

2.3 Mamba Architecture and Covariance Scaling

The temporal backbone uses a Mamba selective state-space block (Gu and Dao, 2023). Mamba has linear sequence complexity, whereas standard self-attention has quadratic sequence complexity. These properties motivate its investigation for GNSS temporal modelling but do not independently establish superiority over recurrent or attention-based architectures. Corresponding LSTM and Transformer comparisons are therefore presented in Section 3.

A linear input layer maps the 20 features at each epoch to a 64-dimensional representation. The resulting sequence is processed by a Mamba block with $d_{model} = 64$, $d_{state} = 16$, $d_{conv} = 4$, and an expansion factor of 2. The final sequence representation is mapped to six variance scale factors:

$$\alpha = \left[\alpha_{Code}^{GPS}, \alpha_{Phase}^{GPS}, \alpha_{Code}^{GLO}, \alpha_{Phase}^{GLO}, \alpha_{Code}^{GAL}, \alpha_{Phase}^{GAL}\right] \#(1)$$

A Softplus transformation ensures positive outputs, and the resulting values are constrained to $[0.1, 10]$. Values greater than one increase a measurement variance, whereas values below one decrease it.

During positioning, each predicted factor multiplies the corresponding variance produced by the existing C/N□-weighted RTKLIB stochastic model:

$$R_{k,t}^{\text{adaptive}} = \alpha_{g(k),t} R_{k,t}^{\text{RTKLIB}} \#(2)$$

where k denotes an individual measurement and g(k) maps that measurement to one of the six constellation and observation-type groups. Consequently, the proposed model adds temporal, group-level adaptation on top of the existing RTKLIB C/N_0-dependent measurement weighting rather than replacing the underlying stochastic model.

During model training, reference variances of $\sigma^2_{\text{ref,code}} = 1\ \text{m}^2$ and $\sigma^2_{\text{ref,phase}} = 0.0001\ \text{m}^2$ are used to normalize the code and phase residual contributions in the loss. These reference quantities are distinct from the C/N□-dependent variances generated by RTKLIB and scaled during final positioning.

The six-factor formulation provides a fixed output representation despite changes in satellite visibility. However, all measurements within the same constellation and observation-type group receive a common scale factor. Satellite counts affect the contribution of each group to the training loss but do not produce satellite-specific scale factors. Consequently, individual degraded and reliable satellites within the same group cannot be reweighted independently. Per-satellite adaptive scaling is therefore identified as future work.

2.4 Loss Function

The model is trained using a composite objective consisting of Gaussian negative log-likelihood, posterior variance-factor regularization, stability regularization, and temporal smoothness.

$$L_{NLL} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{v_i^2}{\alpha_i \sigma^2_{ref,i}} + \log\left(\alpha_i \sigma^2_{ref,i}\right)\right) \#(3)$$

where v_i denotes a ground-truth-referenced residual for measurement channel i, α_i is its predicted variance scale factor, $\sigma^2_{\text{ref,i}}$ is the corresponding training reference variance, and N is the number of residual channels considered.

A posterior variance-factor inspired term is used to enforce stochastic consistency (χ^2). In the implemented loss, the posterior variance factor is explicitly encouraged to remain close to 1, since a value near unity indicates that the predicted stochastic model is consistent with the observed residual behavior.

$$\sigma_0^2 = \frac{v^T P v}{n-u} \#(4)$$

where σ_0^2 is the posterior variance factor. Then the loss factor is determined as:

$$L_{VF} = mean((\sigma_0^2 - 1)^2)\ \#(5)$$

where v is the residual vector, P contains the inverse predicted variances, n is the number of observations, and u = 4 is the number of unknowns corresponding to the three-dimensional receiver position and receiver clock bias. Both code and phase residual groups are included. To account for unequal observation availability, each constellation- and observation-type-level contribution is weighted by its corresponding number of available satellites at that epoch. This count-based weighting affects the contribution of each group to the training loss; it does not provide satellite-specific alpha predictions.

To prevent unnecessary variance inflation, an L1 stability regularization term penalizes deviations from the nominal hypothesis $\alpha = 1$.

$$L_{stability} = mean(|\alpha_t - 1|) \#(6)$$

A temporal smoothness constraint is also introduced using the Huber loss between two consecutive alpha predictions:

$$L_{smoothness} = smooth_{L1}\left(\alpha_t, \alpha_{\{t-1\}}\right) \#(7)$$

The combined loss function is computed as follows:

$$L_{total} = (w_{NLL} * L_{NLL}) + (w_{VF} * L_{VF}) + \left(w_{stability} * L_{stability}\right) + (w_{smoothness} * L_{smoothness}) \#(8)$$

The selected weights are 1.0, 0.6, 0.08, and 0.03, respectively, based on the tuning experiments reported in Section 3.

2.5 Training and Inference

The model is optimized using Adam, with gradient clipping applied for numerical stability. The temporal smoothness term is calculated only between adjacent windows belonging to the same drive-phone trajectory.

During inference, the preprocessing transformations fitted on the training set are applied unchanged to the test features. Each predicted alpha vector is aligned with its assigned timestamp and supplied to RTKLIB. RTKLIB is then rerun with the adapted measurement covariance matrix R, and the resulting positions are converted into the submission file used for GSDC leaderboard evaluation. The entire procedure is performed offline on the experimental workstation and does not represent real-time on-device execution.

3 Experiments and Results

This section evaluates the proposed Mamba-AKF framework from four perspectives: positioning accuracy, architectural comparison, temporal-context selection, and computational efficiency. The primary evaluation uses the GSDC 2022 test trajectories and the competition scoring metric, with the C/N□ weighted RTKLIB solution serving as the conventional stochastic-model baseline. Additional analysis is conducted on the training trajectories, where ground truth is available, to examine

mean, median, RMS, P95, and maximum horizontal positioning errors. The following subsections first report this benchmark result followed by controlled architecture comparison, trajectory-level error analysis, ablation studies, computational evaluation, and interpretation of the predicted covariance scale factors.

3.1 Final GSDC 2022 Benchmark Performance

The primary evaluation of the proposed framework was conducted on the GSDC 2022 test trajectories using the Kaggle private-leaderboard metric. The trained Mamba model predicted six covariance scale factors for each trajectory, which were applied to the corresponding code and carrier-phase variances of the C/N□-weighted RTKLIB stochastic model. Reprocessing the test trajectories with the resulting adaptive measurement covariance matrix produced a final score of 1.533 m. When compared retrospectively with the final GSDC 2022 private leaderboard, this result would correspond to third position. As the evaluation was performed after the official competition period, it is reported as a retrospective leaderboard-equivalent position rather than an official competition placement. Nevertheless, the result constitutes the main test-set evaluation of this study and demonstrates that the proposed Mamba-AKF framework achieves highly competitive positioning performance on the GSDC 2022 benchmark.

3.2 Architecture Comparison

To evaluate whether the positioning performance was specific to the Mamba backbone, controlled experiments were conducted using LSTM and Transformer architectures. In each experiment, only the temporal backbone was changed. All models used the same 20 input features, 14-epoch centered window, six constellation- and observation-type-specific alpha outputs, composite loss formulation, and training/test datasets. After training, each architecture predicted alpha values for the GSDC 2022 test trajectories. The predicted values were then applied to the same C/N₀ -weighted RTKLIB covariance model, and the resulting positions were evaluated through the Kaggle private leaderboard.

Table 2. GSDC 2022 private-leaderboard scores obtained using Mamba, Transformer, and LSTM

Temporal backbone	GSDC Score (m)
Mamba	**1.533**
Transformer	1.608
LSTM	1.626

Under this common experimental configuration, Mamba achieved the lowest positioning score. It improved upon the Transformer and LSTM results by 0.075 m and 0.093 m, respectively, corresponding to relative score reductions of approximately 4.7% and 5.7%. The results indicate that the choice of temporal backbone affects the learned covariance scale factors and their downstream positioning performance. For the evaluated 20-feature, 14-epoch configuration, Mamba provided the most effective alpha predictions among the three tested architectures.

This experiment is intended as a controlled comparison under the implemented settings rather than an exhaustive assessment of each architecture. Architecture-specific hyperparameter optimization could potentially improve the LSTM or Transformer results; therefore, the findings support Mamba's effectiveness in this framework but do not establish universal superiority over recurrent or attention-based models.

3.3 Training-Trajectory Error Analysis

To further examine the positioning behavior of the proposed framework, an analysis was conducted on the 109 training drive-phone trajectories for which ground-truth coordinates were available. Based on trajectory-level RMS horizontal error, Mamba-AKF outperformed the C/N₀ -weighted RTKLIB baseline on 59 trajectories, corresponding to 54.1% of the evaluated cases. The following representative trajectories illustrate how the adaptive covariance scaling affects the epoch-wise 2D horizontal positioning error, including reductions in sustained errors and larger error excursions.

Fig. 1. presents the trajectory with the largest RMS-error reduction among the evaluated training cases. For the highway-suburban trajectory, 2021-12-07-US-LAX-1/ SamsungGalaxyS20Ultra, Mamba-AKF reduced the RMS horizontal error from 1.142 m to 0.852 m, corresponding to a 25.4% reduction. Consistent improvements were also obtained in the mean, median, P95, and maximum errors as shown in Table 3.

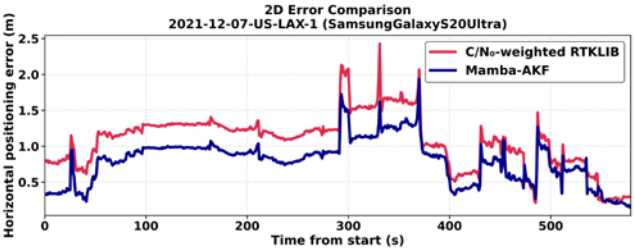


Fig. 1. Epoch-wise horizontal positioning errors for the 2021-12-07-US-LAX-1 SamsungGalaxyS20Ultra training trajectory. Mamba-AKF consistently reduces the error relative to the C/N₀ -weighted RTKLIB baseline.

Table 3. Horizontal Error Statistics for the trajectory 2021-12-07-US-LAX-1/SamsungGalaxyS20Ultra

Statistic (m)	C/N_0-weighted RTKLIB	Mamba-AKF	Improvement (%)
Mean	1.077	0.794	26.309%
Median	1.149	0.852	25.861%
RMS error	1.142	0.852	25.408%
P95 error	1.648	1.287	21.925%
Maximum error	2.430	1.937	20.300%

The second representative trajectory 2021-04-08-US-MTV-1/SamsungGalaxyS20Ultra covers a short transition from a suburban street environment to a wider highway-like roadway while remaining within a predominantly suburban setting. Across 135 matched epochs, Mamba-AKF consistently outperformed the C/N□ -weighted RTKLIB baseline: the mean error decreased by 22.8%, the median by 20.7%, and the RMS error from 1.359 m to 1.054 m (22.5%). Similar reductions of approximately 21% were observed for P95, and maximum error, indicating consistent improvement across the complete error distribution. Table 4. provides error metrics for the second trajectory.

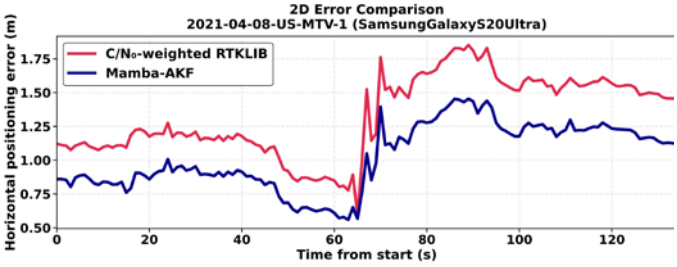


Fig. 2. Epoch-wise horizontal positioning errors for the 2021-04-08-US-MTV-1 SamsungGalaxyS20Ultra training trajectory, covering a transition from suburban streets to a wider highway-like roadway. Mamba-AKF maintains lower errors than the C/N₀ -weighted RTKLIB baseline throughout most of the trajectory.

Table 4. Horizontal Error Statistics for the trajectory 2021-04-08-US-MTV-1/SamsungGalaxyS20Ultra

Statistic (m)	C/N_0-weighted RTKLIB	Mamba-AKF	Improvement (%)
Mean	1.327	1.025	22.754%
Median	1.230	0.975	20.718%
RMS error	1.359	1.054	22.469%
P95 error	1.778	1.407	20.832%
Maximum error	1.852	1.453	21.501%

The third representative case demonstrates performance over a substantially longer highway-suburban trajectory 2020-05-29-US-MTV-2 / GooglePixel4. Mamba-AKF reduced the mean error from 0.850 m to 0.727 m and the RMS error from 0.976 m to 0.803 m, corresponding to improvements of 14.5% and 17.7%, respectively. P95 was reduced by 17.0%, while the median remained nearly unchanged. The maximum Mamba-AKF error was marginally higher by 0.128 m at the trajectory endpoint. Map and velocity inspection indicate that this epoch coincided with a terminal turning or acceleration event following a low-speed interval. As the maximum represents a single epoch, the broader mean, RMS, and P95 statistics in Table 5. provides stronger evidence of the overall improvement.

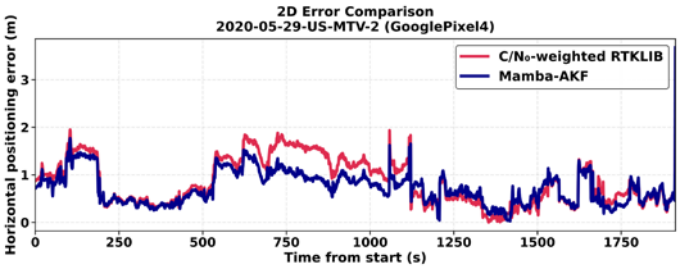


Fig. 3. Epoch-wise horizontal positioning errors for the 2020-05-29-US-MTV-2 GooglePixel4 training trajectory. Mamba-AKF reduces sustained and upper-percentile errors over most of the trajectory, while a localized endpoint event produces a slightly larger maximum error.

Table 5. Horizontal Error Statistics for the trajectory 2020-05-29-US-MTV-2/GooglePixel4

Statistic (m)	C/N_0-weighted RTKLIB	Mamba-AKF	Improvement (%)
Mean	0.850	0.727	14.452%
Median	0.695	0.687	1.159%
RMS error	0.976	0.803	17.743%
P95 error	1.643	1.364	16.965%
Maximum error	3.550	3.678	-3.605%

The fourth trajectory 2021-01-04-US-SFO-1/XiaomiMi8 represents a predominantly highway environment with surrounding tree cover and contains 1,986 epochs. The C/N□ -weighted RTKLIB and Mamba-AKF solutions remain similar over many nominal intervals, but Mamba-AKF provides consistent reductions in the overall error distribution. Mean, median, and RMS errors decreased by 9.3%, 9.9%, and 10.1%, respectively, while P95 decreased by 11.6%. The maximum error was also reduced from 9.389 m to 7.732 m, corresponding to a 17.6% reduction. This case demonstrates that the learned covariance scaling can provide moderate improvements without substantially altering already stable positioning intervals. Fig. 4. presents the horizontal positioning errors for this representative trajectory, while Table 6. summarizes their corresponding error statistics.

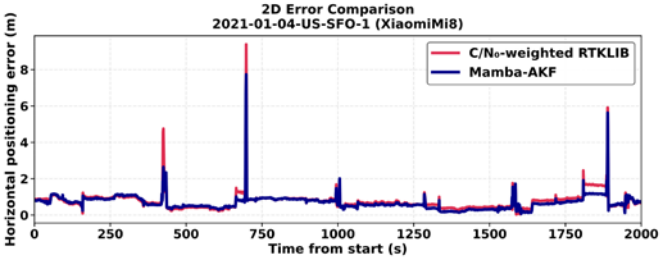


Fig. 4. Epoch-wise horizontal positioning errors for the 2021-01-04-US-SFO-1 XiaomiMi8 training trajectory in a tree-lined highway environment. The two solutions remain similar during nominal intervals, while Mamba-AKF moderately reduces sustained and extreme errors.

Table 6. Horizontal Error Statistics for the trajectory 2021-01-04-US-SFO-1/XiaomiMi8

Statistic (m)	C/N_0-weighted RTKLIB	Mamba-AKF	Improvement (%)
Mean	0.744	0.675	9.329%
Median	0.699	0.630	9.922%
RMS error	0.859	0.772	10.086%
P95 error	1.294	1.144	11.591%
Maximum error	9.389	7.732	17.645%

3.4 Ablation and Temporal-Context Analysis

Ablation experiments were conducted to select the input features, loss configuration, and temporal-window length. The initial 22-feature representation included HDOP and VDOP. Removing these two variables improved the benchmark result, whereas additional removal of code-residual or state-uncertainty features degraded performance. The final representation therefore retained the residual, signal-strength, satellite-availability, kinematic, state-uncertainty, and device-identity features, resulting in the 20-feature configuration described in Section 2.

3.4.1 Loss-Weight and Variance-Factor Ablation

Loss-weight tuning was initially performed using a 16-epoch window. Table 7. summarizes the evaluated combinations of NLL, variance-factor, stability, and smoothness weights. Among these configurations, $w_{NLL} = 1.0$, $w_{VF} = 0.6$, $w_{stability} = 0.08$, and $w_{smoothness} = 0.03$ achieved the lowest score of 1.573 m. This configuration was retained during the subsequent window-length experiments.

Table 7. Loss-weight configurations evaluated using a 16-epoch temporal window.

w_{NLL}	w_{VF}	$w_{stability}$	$w_{smoothness}$	GSDC Score (m)
1.0	0.3	0.08	0.03	1.661
1.0	0.5	0.02	0.03	1.643
1.0	0.5	0.05	0.03	1.635
1.0	0.5	0.05	0.02	1.647
1.0	0.5	0.08	0.03	1.608
1.0	0.6	0.05	0.03	1.618
0.6	0.6	0.08	0.03	1.638
0.7	0.6	0.08	0.03	1.610
0.8	0.6	0.08	0.03	1.604
1.0	**0.6**	**0.08**	**0.03**	**1.573**
1.0	0.6	0.08	0.04	1.644
2.0	0.6	0.08	0.03	1.670
1.0	0.6	0.09	0.03	1.643
1.0	0.7	0.05	0.03	1.654
1.0	0.7	0.08	0.03	1.645

The composition of the variance-factor term was also examined. Code-only and phase-only formulations produced scores of 1.673 m and 1.726 m, respectively. Hybrid code-phase weighting produced scores between 1.625 m and 1.650 m, but none improved upon the 1.573 m obtained when both residual types were included without an additional relative weighting coefficient. The unseparated code-and-phase formulation was therefore retained.

3.4.2 Window-Length Selection and Residual Autocorrelation

After selecting the loss weights, window lengths of 12, 13, 14, 15, and 20 epochs were evaluated using the same feature representation and training objective. The 14-epoch configuration achieved the lowest GSDC score of 1.533 m and was therefore selected for the final model. Shorter windows provided less temporal context, whereas longer windows did not improve the benchmark result. Accordingly, the selected length should be interpreted as an empirical optimum for the evaluated dataset and configurations rather than as a universally optimal GNSS window length.

Table 8. GSDC benchmark scores obtained for the evaluated temporal-window lengths.

Window length	GSDC Score (m)
12	1.641
13	1.622
14	**1.533**
15	1.569
20	1.643

Residual autocorrelation was examined to determine whether the input sequences contained temporal information that could support window-based modelling. Fig. 5. shows the mean ACF of the absolute constellation-level code and carrier-phase residual sequences across the training drive-phone trajectories. The strongest dependence occurs at short lags and gradually decreases as the lag increases. At lag 7, corresponding to the preceding context of the selected centered window, the mean ACF remains positive for all six residual channels and ranges from approximately 0.074 to 0.190. Positive dependence also remains at lag 14, although at a reduced magnitude. These results indicate that residual magnitudes persist across neighboring epochs and support temporal rather than pointwise prediction. However, the ACF does not identify a unique cutoff at 14 epochs; the final window length was selected empirically using the positioning-performance results reported in Table 8.

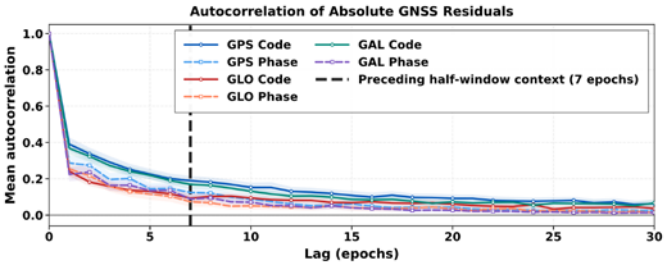


Fig. 5. Mean autocorrelation functions of the absolute constellation-level code and carrier-phase residual sequences across the training drive-phone trajectories. The vertical dashed line at lag 7 indicates the preceding context of the selected centered 14-epoch window.

3.5 Computational Efficiency and Runtime

The computational performance of Mamba, LSTM, and Transformer was measured on an NVIDIA GeForce RTX 4060. All models received input tensors with shape $(128,14,20)$, representing a batch of 128 windows, 14 epochs per window, and 20 input features. The controlled benchmark used the same warm-up and timed iterations for all models, with input tensors already loaded on the GPU. End-to-end alpha prediction was also timed over the 65,303 windows in the complete test dataset.

Mamba had the fewest trainable parameters and the smallest model size, although the differences among the three models were small. LSTM achieved the lowest inference time, processing the complete test set in 0.159 s. Transformer required 0.270 s, while Mamba required 0.466 s. Their corresponding test-set throughputs were approximately 409,909, 242,099, and 140,237 windows/s, respectively. Transformer used the least additional GPU memory, while Mamba used approximately 53% less additional memory than LSTM.

Table 9. Computational performance and positioning scores of the evaluated temporal architectures

Metric	Mamba	LSTM	Transformer
Trainable parameters	34374	35014	36230
Model size (MB)	0.131	0.134	0.138
Batch inference time (ms)	0.589 ± 0.023	0.224 ± 0.004	0.426 ± 0.014
Additional GPU memory (MB)	9.13	19.58	4.79
Full test inference (s)	0.466	0.159	0.270
GSDC score (m)	1.533	1.626	1.608

Although Mamba has linear complexity with respect to sequence length, it was not the fastest model for the short 14-epoch sequences used in this study. Nevertheless, all three models generated alpha predictions for the complete test dataset in less than 0.5 s. Therefore, neural-network inference itself added only a small computational cost to the offline positioning pipeline. Mamba achieved the lowest GSDC positioning score despite having the highest measured inference time. Its main benefit in the present experiment should therefore be interpreted as improved positioning accuracy rather than a runtime advantage.

The reported batch timings measure model execution with inputs already loaded on the GPU. The full test-set timings measure batched alpha prediction but exclude feature extraction, RTKLIB processing, final solution generation, and file writing. The GPU-memory values represent additional memory allocated during inference rather than total GPU or system memory. Because the experiments were performed on a desktop GPU, these results describe workstation-based post-processing and should not be interpreted as measurements of real-time execution on a smartphone.

3.6 Predicted Scale Factors and Input-Feature Traces

To examine the temporal behavior of the learned covariance adaptation, Fig. 6. aligns the six predicted variance scale factors with the corresponding residual, C/N□, satellite-count, and velocity features for a representative test trajectory. The panels use the same time axis, allowing changes in the predicted factors to be compared with the measurement and motion context available to the model.

During the initial part of the trajectory, several large code- and carrier-phase-residual excursions coincide with pronounced increases in the corresponding scale factors. In particular, the elevated GLONASS code residuals are accompanied by larger GLONASS code factors, while localized carrier-phase residual excursions occur near sharp changes in the phase factors. Some outputs reach the imposed upper bound of 10 during these intervals, indicating strong down-weighting of the corresponding observation group. These saturated values should be interpreted as activation of the model's output limit rather than as precise estimates beyond that boundary.

A different behavior is visible near the middle of the trajectory. Several scale factors increase even though the residual features remain comparatively moderate. These changes occur near variations in C/N□, reduced satellite availability, and a transition in receiver velocity. This observation suggests that the model does not determine covariance scaling from residual magnitude alone, but combines residual information with signal strength, observation availability, motion, and the surrounding temporal context. Near the end of the trajectory, increased code-residual activity and changes in C/N□ and receiver motion again coincide with stronger code-scale adjustments.

The responses also differ across constellations and observation types. For example, the GLONASS code factor remains generally higher than the GPS and Galileo code factors over substantial parts of the trajectory, whereas several phase factors remain relatively stable except during localized events. This branch-specific behavior confirms that the model does not apply one

common scaling factor to the entire measurement covariance matrix.

The aligned traces instead provide qualitative evidence that the learned scale factors vary with the combined temporal measurement context. Together with the positioning and architecture comparisons, these results show that the proposed framework performs time-varying, group-level covariance adaptation within the evaluated post-processing pipeline.

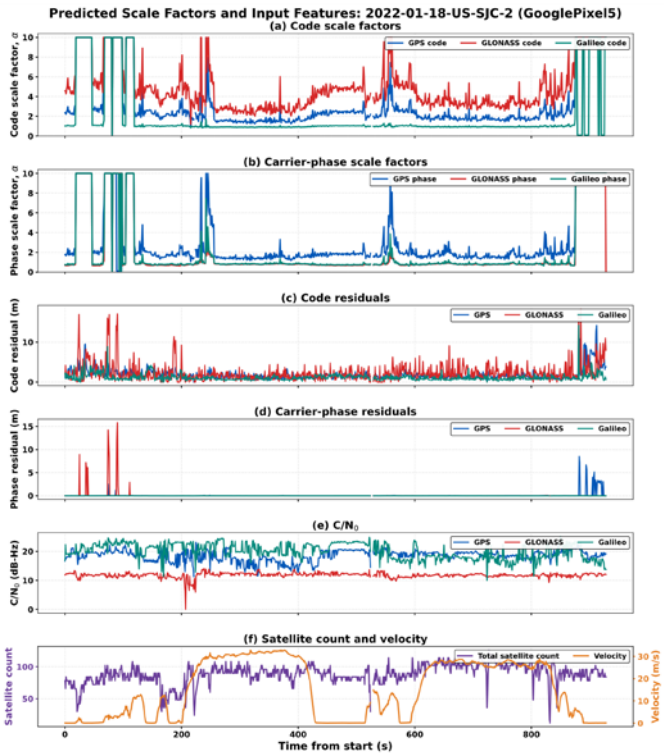


Fig. 6. Predicted variance scale factors and input-feature traces for the representative test trajectory 2022-01-18-US-SJC-2 collected using a Google Pixel 5. The aligned panels show code and carrier-phase scale factors, residuals, C/N$_0$, satellite availability, and receiver velocity. The figure illustrates temporal associations between the model inputs and predicted factors but does not imply a direct causal relationship for any individual feature.

4. Conclusion & Future Work

This study investigated a Mamba-augmented adaptive Kalman filter for post-processed smartphone GNSS positioning. Using a centered 14-epoch window, the model predicts six variance scale factors for the GPS, GLONASS, and Galileo code and carrier-phase observation groups. These factors modify the existing C/N□ weighted RTKLIB measurement variances while leaving the process noise covariance and remaining positioning configuration unchanged.

The final configuration achieved a GSDC 2022 private-leaderboard score of 1.533 m, which would retrospectively correspond to third position on the competition leaderboard. This result was achieved using a GNSS-only, software-based framework without IMU integration, sensor fusion, or additional hardware. Under the matched architecture comparison, Mamba produced lower scores than the Transformer and LSTM models, which obtained 1.608 m and 1.626 m, respectively. This result demonstrates the effectiveness of the Mamba backbone for the evaluated 20-feature, 14-epoch configuration, but it is not intended to establish universal superiority over recurrent or attention-based models. Architecture-specific tuning could potentially change their relative performance.

Analysis of the 109 training drive-phone trajectories showed that Mamba-AKF reduced the trajectory-level RMS horizontal error in 59 cases, corresponding to 54.1% of the evaluated trajectories. The representative cases demonstrated reductions in sustained and upper-percentile errors across different suburban and highway environments, although improvement was not uniform across every trajectory or error statistic. Furthermore, Mamba was not the fastest architecture for the short 14-epoch sequences: LSTM and Transformer completed inference faster. Nevertheless, all three models processed the complete test set in less than 0.5 s on the experimental workstation. Mamba's benefit in this study should therefore be interpreted as improved positioning accuracy under the evaluated configuration rather than as a runtime advantage or evidence of universal architectural superiority.

The present study is limited to offline processing with a non-causal centered window and group-level covariance scaling. It cannot independently adjust individual satellites within the same observation group. Furthermore, the model was trained using ground-truth-referenced residuals and tuned on the GSDC benchmark, while comparison with conventional methods was limited to the implemented C/N□-weighted RTKLIB baseline. Generalization to other datasets, devices, and processing configurations therefore remains to be evaluated.

Future work will examine satellite-specific scaling, causal inference for real-time operation, and validation on independent datasets and mobile hardware. Controlled comparisons with elevation- and C/N$_0$ -based stochastic models, innovation-based adaptive estimation, Sage–Husa filtering, covariance matching, and LS-VCE are also required. Hybrid approaches that combine learned temporal representations with statistically grounded variance estimation may provide a useful direction for improving robustness while preserving interpretability.

REFERENCES

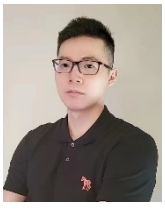
1. Banville, S., & van Diggelen, F. (2016). Precise GNSS for everyone: Precise positioning using raw GPS measurements from android smartphones. GPS World, 27(11), 43–48.

2. Groves, P., Jiang, Z., Rudi, M., & Strode, P. (2013). A portfolio approach to NLOS and multipath mitigation in dense urban areas. Proceedings of the 26th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2013), 3231–3247.

https://www.ion.org/publications/abstract.cfm?articleID=11264
3. Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2312.00752
4. Howard, A., Chow, A., Julian, B., Orendorff, D., Fu, M., Khider, M., & Dane, S. (2022). Google smartphone decimeter challenge 2022. Kaggle. https://kaggle.com/competitions/smartphone-decimeter-2022
5. Hsu, L.-T. (2017). GNSS multipath detection using a machine learning approach. 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), 1–6. https://doi.org/10.1109/ITSC.2017.8317700
6. Jiang, Y., Zhou, Z., Zhang, Y., Yang, H., & Gao, Y. (2025). Deep-learning-enhanced outlier detection for precise GNSS positioning with smartphones. IEEE Transactions on Instrumentation and Measurement, 74, 1–13. https://doi.org/10.1109/TIM.2025.3562977
7. Li, Y., Cai, C., & Xu, Z. (2022). A combined elevation angle and C/N0 weighting method for GNSS PPP on Xiaomi MI8 Smartphones. Sensors, 22(7), 2804. https://doi.org/10.3390/s22072804
8. Li, Z., Zeng, K., Wang, L., Kan, X., Yuan, R., & Xie, S. (2023). NLOS/LOS classification by constructing indirect environment interaction from GNSS receiver measurements using a transformer-based deep learning model. Proceedings of ION ITM 2023, 859–870. https://doi.org/10.33012/2023.18657
9. Nie, S., & Yang, H. (2025). Attention-GANs: An advanced GNSS data augmentation method for improved NLOS/LOS classification. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLVIII-G-2025, 1117–1122. https://doi.org/10.5194/isprs-archives-XLVIII-G-2025-1117-2025
10. Nix, D. A., & Weigend, A. S. (1994). Estimating the mean and variance of the target probability distribution. Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), 1, 55–60. https://doi.org/10.1109/ICNN.1994.374138
11. Paziewski, J., Sieradzki, R., & Baryla, R. (2019). Signal characterization and assessment of code GNSS positioning with low-power consumption smartphones. GPS Solutions, 23(4), 98. https://doi.org/10.1007/s10291-019-0892-5
12. Shinghal, G., & Bisnath, S. (2021). Conditioning and PPP processing of smartphone GNSS measurements in realistic environments. Satellite Navigation, 2(1), 10. https://doi.org/10.1186/s43020-021-00042-2
13. Suzuki, T., & Amano, Y. (2021). NLOS multipath classification of GNSS signal correlation output using machine learning. Sensors, 21(7), 2503. https://doi.org/10.3390/s21072503
14. Teunissen, P. J. G., & Amiri-Simkooei, A. R. (2008). Least-squares variance component estimation. Journal of Geodesy, 82(2), 65–82. https://doi.org/10.1007/s00190-007-0157-x
15. Vishwanath, A., Mansell, T., Nie, S., Wuran, D., & Yang, H. (2026). Pseudorange generation in Android smartphones: Converter inconsistencies, root-cause analysis, and positioning effects. Proceedings of the ION ITM 2026, 576–589. https://doi.org/10.33012/2026.20567
16. Yin, X., Chai, H., Xiang, M., Du, Z., & Tian, X. (2021). A GEO-pivoted adaptive extended Kalman filtering method in low-latitude dense building environment. Measurement, 173, 108657. https://doi.org/10.1016/j.measurement.2020.108657
17. Zangenehnejad, F., & Gao, Y. (2021). GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. Satellite Navigation, 2(1), 24. https://doi.org/10.1186/s43020-021-00054-y
18. Zangenehnejad, F., & Gao, Y. (2023). Stochastic modeling of smartphones GNSS observations using LS-VCE and application to Samsung S20. Sensors, 23(7), 3478. https://doi.org/10.3390/s23073478
19. Zhang, Z., Li, B., Shen, Y., Gao, Y., & Wang, M. (2018). Site-specific unmodeled error mitigation for GNSS positioning in urban environments using a real-time adaptive weighting model. Remote Sensing, 10(7), 1157. https://doi.org/10.3390/rs10071157
20. Zhang, Z., Li, Y., He, X., Chen, W., & Li, B. (2022). A composite stochastic model considering the terrain topography for real-time GNSS monitoring in canyon environments. Journal of Geodesy, 96(10), 79. https://doi.org/10.1007/s00190-022-01660-7
21. Zhou, Z., Stefanakis, D., Liu, B., & Yang, H. (2023). Hopular-based GNSS signal reception classification method for LOS/NLOS detection in urban environments. Proceedings of ION GNSS+ 2023, 2990–3004. https://doi.org/10.33012/2023.19358



Divya Bhavsar is a Master of Science (MSc) student in Geomatics Engineering at the University of Calgary, Canada. He holds a Bachelor's degree in Computer Engineering and his research interests include GNSS positioning, machine learning for navigation, and software development.



Dr. Hongzhou Yang has been an Assistant Professor in the Department of Geomatics Engineering at the University of Calgary, Canada since 2022 and serves as the Director of Intelligent Navigation and Mapping Lab. His research focuses on localization and mapping, estimation theory, machine learning for navigation, navigation in challenging environments, intelligent transportation systems and self-driving.