

Development of Intelligent 3D UAV Navigation Based on Inertial and Satellite Data

Ayan Kemel^{1,*}, Sabyrzhan Atanov¹, and Luigi La Spada²

¹Department of Computer and Software Engineering, L.N. Gumilyov Eurasian National University, Satpayev str. 2, Astana 010008, Republic of Kazakhstan

²School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, United Kingdom

Received: 2 Jul 2026, Revised: 15 Sep. 2026, Accepted: 22 Oct. 2026

Published online: 1 Nov. 2026

Abstract: Reliable three-dimensional navigation of unmanned aerial vehicles requires continuity when satellite positioning is temporarily unavailable. This paper presents Adaptive FusionNav, a causal recurrent model that combines inertial measurements with causally available Global Navigation Satellite System (GNSS) history and learns a bounded velocity attenuation gate for GNSS outages. Experiments use the real-flight INSANE dataset with flight-level separation into training, validation, and previously unused test sequences. Artificial GNSS outages of 10, 30, and 60 s are evaluated with three neural-network seeds. On held-out test flights, the IMU-enabled model achieved relative-motion RMSEs of 1.356, 8.972, and 17.222 m, compared with 2.055, 9.908, and 19.325 m for a parameter-matched GNSS-only neural model, corresponding to reductions of 34.0%, 9.4%, and 10.9%. The IMU-enabled variant was better in 17 of 18 test episode–seed comparisons. However, simple Held-GNSS and kinematic baselines remained stronger on the aggregate held-out test. The results therefore support a reproducible incremental contribution of inertial data within the matched learned fusion architecture, rather than universal superiority over classical baselines.

Keywords: UAV navigation; 3D navigation; inertial measurement unit; GNSS; sensor fusion; GNSS outage; recurrent neural network; adaptive fusion

1 Introduction

Accurate and continuous positioning is a prerequisite for autonomous unmanned aerial vehicle (UAV) operation. Global Navigation Satellite Systems (GNSS) provide a convenient absolute reference outdoors, but the availability and quality of satellite measurements can deteriorate in urban canyons, near large structures, under intentional interference, or during transitions between outdoor and enclosed environments. Recent reviews of GNSS-denied UAV navigation emphasize that robust localization remains an open system-level problem involving sensor availability, computational cost, platform constraints, and the complementary characteristics of absolute and relative navigation sources [1,2]. At the propagation level, Ioannidi et al. [3] analyzed radiation from a vertical Hertzian dipole above lossy ground using a spectral-domain formulation. In parallel, broad reviews of GNSS/inertial navigation system (INS) integration show that satellite and inertial sensors remain the core pair for continuous navigation because the long-term

bounded character of GNSS errors complements the high-rate but drifting nature of inertial mechanization [4, 5]. Trajectory-estimation literature further shows that the practical value of a fusion method depends not only on estimator structure but also on calibration, synchronization, reference frames, and the evaluation protocol used to quantify drift [6]. A related multi-robot setting is formation control and swarm coordination, investigated by Kyzrkanov et al. [7].

Conventional GNSS/INS integration is commonly formulated with Kalman-filter variants or factor-graph estimation. These approaches have strong physical interpretability, but their performance depends on a sufficiently complete sensor model and on calibrated quantities such as biases, noise levels, sensor lever arms, timing, and initial attitude. Low-cost micro aerial vehicles complicate this picture because aggressive dynamics, vibration, limited size-weight-power budgets, and short missions interact with MEMS sensor quality. Classical adaptive methods have therefore been proposed for

* Corresponding author e-mail: kemel.am.1@enu.kz

commercial UAV flight-control systems, including noise adaptation based on GNSS quality [8]. At the same time, recent surveys document rapidly growing interest in machine learning methods for inertial sensing, GNSS quality enhancement, and multisensor fusion [9,10]. Data-driven models are attractive because they can represent temporal dependencies and nonlinear relationships without requiring all stochastic sensor effects to be specified analytically. Their flexibility, however, creates a second challenge: a neural network can exploit easy static correlations, timing artifacts, or information leakage rather than the intended sensor dynamics if the experiment is not carefully controlled.

Learning-based outage bridging has consequently evolved from static multilayer perceptrons to recurrent and convolutional-recurrent models. Fang et al. used long short-term memory (LSTM) units to estimate pseudo-measurements during GNSS signal loss [11]; Zhi et al. combined convolutional and recurrent processing for GPS/INS compensation [12]; Tang et al. coupled a gated recurrent unit (GRU) predictor with an adaptive Kalman filter [13]; and Zhao et al. proposed CNN-GRU assistance with a cubature Kalman filter [14]. More recent studies have considered cascaded neural predictors [15], lightweight odometry networks [16], transformer-like temporal processing [17], Gaussian-process regression [18], cascade-LSTM pseudo-velocity measurements [19], LSTM-factor-graph hybrids [20], and deep error-compensation networks [21]. These methods collectively demonstrate the usefulness of learned temporal models, but direct comparison across papers is difficult because the sensors, platforms, outage construction, reference trajectories, test protocols, and performance metrics differ substantially.

A related body of work studies learned inertial odometry independently of GNSS. IONet [22], RIDI [23], RoNIN [24], TLIO [25], and AI-IMU dead reckoning [26] show several ways of embedding learned priors into inertial motion estimation. Visual-inertial research has also established rigorous treatments of causal sensor fusion and inertial integration, including nonlinear keyframe optimization [27], on-manifold preintegration [28], VINS-Mono [29], and learned visual-inertial sequence models such as VINet [30]. These works are not interchangeable with GNSS/IMU navigation, but they highlight a recurring principle: learned components are most informative when the experiment isolates the information contributed by each sensing modality.

This paper adopts that principle. Rather than asking only whether a neural trajectory predictor has a low error, we construct a matched sensor-ablation experiment in which the proposed model and an IMU-ablated GNSS/context-only counterpart have the same architecture, parameter count, non-IMU context, training order, and evaluation protocol. The sole systematic difference is whether the six normalized acceleration and angular-rate values are available to the recurrent fusion network. This controlled comparison is important because

the absence of such matching can make improvements attributable to architecture size, optimization, or GNSS-history processing appear to be benefits of inertial information.

The experiments use the INSANE dataset [31], which provides real UAV flights, multiple inertial and satellite sensors, calibration information, and high-quality outdoor reference trajectories. The flight sequences are separated by complete flights: `mars_1--mars_3` are used for training, `mars_4--mars_5` for validation and checkpoint selection, and `mars_6--mars_7` as a held-out test that is not used for model development. We simulate complete ordinary-GNSS outages of 10, 30, and 60 s after fixing all masks and evaluation rules. The learned model is compared with a parameter-matched IMU-ablated GNSS/context-only neural predictor and three transparent kinematic baselines: holding the last available GNSS position, causal constant-velocity extrapolation, and a train-calibrated damped constant-velocity model.

The study makes four contributions: (i) a compact causal adaptive recurrent architecture for combining IMU measurements with GNSS history during satellite outages; (ii) a leakage-controlled, flight-separated protocol with a parameter-matched IMU ablation and three random seeds; (iii) a previously unused held-out test that confirms a repeatable reduction in error when inertial measurements are enabled inside the matched model; and (iv) a transparent comparison against simple kinematic baselines that establishes the practical scope of the learned predictor. The resulting evidence isolates a reproducible inertial contribution without relying on changes in model capacity, timing context, or GNSS-history processing.

The experimental design is organized around three research questions. **RQ1** asks whether the IMU-enabled Adaptive FusionNav has lower outage motion error than the parameter-matched GNSS-only network when all non-IMU information is held constant. **RQ2** asks whether any such IMU benefit persists on previously unused flights after validation-based checkpoint selection has been frozen. **RQ3** asks how the learned predictor compares with transparent kinematic rules that require no neural training. These questions deliberately separate sensor contribution from overall method ranking. A method can answer RQ1 positively while still being inferior to a simple baseline under RQ3; the final results show precisely this situation.

The remainder of the paper is organized as follows. Section 2 summarizes related work in classical and learned navigation. Section 3 describes the dataset, causal preprocessing, adaptive fusion architecture, outage protocol, training procedure, baselines, and metrics. Section 4 reports validation and held-out test results. Section 5 discusses what the matched ablation establishes, what it does not establish, and the practical limitations of the current study. Section 6 concludes the paper.

2 Related Work

2.1 Classical and hybrid satellite–inertial navigation

GNSS and inertial navigation have complementary failure modes. GNSS supplies a global position reference whose errors do not grow through integration, while an IMU provides high-rate local motion information but accumulates bias, scale, alignment, and integration errors. The literature therefore contains loosely coupled, tightly coupled, and deeply coupled architectures, with Kalman filtering remaining a central estimator family [4,5]. For UAVs, Zhang and Hsu combined adaptive GNSS-quality modeling with filtering in a commercial flight-control context [8]. Such work illustrates an important point for the present study: even a conceptually simple INS/GNSS filter requires credible initial attitude, noise models, and inter-sensor geometry. When these quantities are unavailable, a diverging filter is not a meaningful baseline against which to claim neural superiority. A complementary system-level problem is movement coordination in multiagent robotic systems, studied by Kzyrkanov et al. [32].

Trajectory estimation can also be formulated by smoothing and factor-graph techniques. Pöpl et al. review integrated trajectory estimation across GNSS, INS, cameras, and mapping sensors [6]. In visual-inertial navigation, keyframe optimization [27], IMU preintegration [28], and systems such as VINS-Mono [29] demonstrate how calibration and temporal constraints can be explicitly represented. Multi-tier UAV localization systems have similarly combined GNSS, inertial, visual, and depth sensing for changing operating conditions [33], while lidar-inertial solutions address GNSS-denied flight in structured environments [34]. Kereyev et al. also investigated satellite-independent indoor positioning with Bluetooth beacons using received-signal-strength filtering and smoothing [35]. These systems can be highly accurate, but they rely on additional exteroceptive sensors or a more complete physical calibration than the two-stream IMU–GNSS setting considered here.

2.2 Learning to bridge GNSS outages

A large group of recent methods treats GNSS interruption as a time-series prediction or error-compensation problem. Fang et al. proposed LSTM-based pseudo-measurements during GNSS outages, explicitly exploiting temporal vehicle dynamics [11]. Zhi et al. used CNN-LSTM processing to compensate GPS/INS degradation with low-cost inertial measurements [12]. Tang et al. used a GRU to predict position variation and combined it with adaptive Kalman filtering [13]. Zhao et al. combined CNN-GRU prediction with a cubature Kalman filter and a real-time training strategy [14]. These

recurrent approaches are conceptually close to the present work in that they exploit sequential dependence, but most are evaluated on road vehicles and differ in what is predicted, how the filter is used, and how outages are constructed.

Subsequent studies have expanded the model family. Cascaded neural networks have been used to model navigation errors during GPS outages [15]. Yu et al. designed a lightweight odometry network using temporal convolution and attention-related mechanisms for GNSS/INS outage support [16]. Kuang and Yan introduced an FECAM-iTransformer architecture for INS/GNSS navigation under signal interruption [17]. Zhu et al. studied improved Gaussian-process regression for pseudo-GNSS generation [18]. Wang et al. proposed cascade-LSTM pseudo-velocity measurements for low-cost GNSS/INS integration [19], while Liu et al. combined LSTM prediction with factor-graph localization [20]. Wu et al. proposed a deep error-compensation approach that combines learning with model-based navigation concepts [21]. PSO-LSTM optimization has also been investigated for airborne-motion error compensation [36]. Taken together, these papers show sustained interest in learning-assisted navigation during GNSS denial, but they also illustrate a methodological risk: when the learned model receives GNSS history, its performance improvement can arise from better extrapolation of GNSS-derived motion even if the inertial branch is weakly used.

The proposed experimental design directly addresses that risk. The GNSS-only ablation is not a different architecture and does not lose time, quality, or availability features. Instead, it zeros only the six normalized accelerometer/gyroscope values while preserving the same recurrent network and parameter count. Therefore, the difference between the two learned variants provides a more focused estimate of the additional predictive value of the IMU within this architecture.

2.3 Learned inertial navigation and multimodal fusion

Pure or predominantly inertial learning provides complementary evidence about the information contained in IMU sequences. IONet formulated inertial odometry as sequential learning to avoid direct double integration of noisy low-cost signals [22]. RIDI learned corrections for robust IMU double integration [23], and RoNIN provided a large benchmark and neural methods for inertial navigation in natural motion [24]. TLIO regressed displacement and uncertainty and fused learned inertial constraints within an EKF-style estimator [25]. Brossard et al. used learned components to adapt an inertial dead-reckoning filter for wheeled vehicles [26]. These studies differ from UAV flight and from the present satellite-aided setting, but they support the general

strategy of learning motion information from inertial sequences instead of treating raw acceleration as a directly integrable position signal.

Deep sensor fusion has also been investigated more generally in autonomous systems. Fayyad et al. review feature-level and decision-level deep sensor fusion for perception and localization [37]. VINet provides an example of learned sequence-to-sequence visual-inertial fusion [30]. The broader lesson is that architecture design alone is insufficient to establish sensor utility: the contribution of an individual modality should be tested through controlled ablations, independent sequences, and clear metrics. This motivates our matched full-versus-GNSS-only experiment.

2.4 UAV datasets, evaluation, and methodological context

The choice of benchmark is particularly important for UAV navigation. INSANE was designed to expose state-estimation algorithms to realistic sensor effects across indoor, outdoor, and Mars-analog scenarios and provides extensive sensor streams plus high-quality reference data [31]. Its outdoor ground truth is based on dual RTK-GNSS and magnetometer geometry and is provided at the PX4 IMU reference. We use only ordinary GNSS and PX4 IMU as model inputs, while reserving the higher-grade reference exclusively for targets and evaluation. This separation supports a stronger experiment than one in which the same GNSS coordinates are used as both model input and nominal ground truth.

Contemporary UAV reviews emphasize that evaluation should account for GNSS-denied localization, multi-sensor trade-offs, computational constraints, and the diversity of platform motion [1,2]. The present work therefore reports not only aggregate means but also per-flight and per-episode outcomes, seed variability, recovery behavior, and cases in which transparent baselines outperform the learned predictor.

For embedded software portability, Brimzhanova et al. [38] described cross-platform compilation of Go programs for Raspberry Pi. Broader computational research also demonstrates the versatility of recurrent learning, few-shot methods, sequential recognition, evolutionary optimization, fraud detection, distributed recommendation, and data-driven information systems across diverse domains [39,40,41,42,43,44,45]. These cross-domain studies provide a wider computational context, while the navigation-specific claims of this work are established by the dedicated navigation literature and the controlled INSANE experiments described below.

3 Materials and Methods

3.1 Dataset and flight-level split

The experiments use the INSANE dataset introduced by Brommer et al. [31]. It contains real micro-aerial-vehicle flights and a broad sensor suite designed for state-estimation research. The present work intentionally restricts the learned input to two sensing families: the published PX4 inertial data and ordinary PX4 GNSS. Acceleration is interpreted in m/s^2 and angular rate in rad/s according to the dataset export. Ordinary GNSS contributes causally available position history, covariance/quality information, availability, and time since the last available update. Ground truth, RTK measurements, magnetometer data, onboard pose, and reference attitude are not provided to the learned encoder or used to reset model state.

The outdoor reference trajectory supplied by the dataset authors is used only for training targets and scoring. The reference positions correspond to the PX4 IMU and are derived from the authors' dual-RTK/magnetometer setup. As documented by the dataset authors, timing and calibration information are part of the published data products [31]. The reference should nevertheless not be interpreted as error-free. In particular, the filename `ground_truth_8hz.csv` does not imply that every row is an independent physical observation at exactly 8 Hz. For this reason, evaluation is performed at the original reference rows and timestamps, without treating an interpolated dense trajectory as additional independent ground-truth samples.

Complete flights, rather than randomly shuffled windows, define the experimental split. The split is frozen before the adaptive experiment and is shown in Table 1. Flights `mars_1`, `mars_2`, and `mars_3` are used for training. Flights `mars_4` and `mars_5` form the validation set used for early stopping and checkpoint selection. Flights `mars_6` and `mars_7` constitute the held-out test set. The test flights were not used for training, validation, checkpoint selection, hyperparameter tuning, or metric-driven model development; they were evaluated only after the architecture, scalars, checkpoints, outage rules, metrics, and damping parameter had been frozen. An initial final-test execution terminated during a pre-inference integrity check; the subsequent retry changed only the numerical tolerance of a non-scientific manifest field, while scenario masks, timestamps, hashes, checkpoints, and all scientific settings remained frozen.

3.2 Causal synchronization and available GNSS history

The preprocessing pipeline is explicitly causal with respect to the information available at each prediction time. Native IMU data, approximately 200 Hz in the

Table 1: Locked experimental protocol used for the final INSANE study.

Item	Locked setting
Training flights	mars_1, mars_2, mars_3
Validation flights	mars_4, mars_5
Held-out test flights	mars_6, mars_7
Learned inputs	PX4 IMU acceleration/gyro; ordinary GNSS displacement/velocity; shared timing, availability, covariance/quality context
Excluded learned inputs	RTK, ground-truth position/attitude, magnetometer, onboard pose
IMU representation	Causal time-weighted aggregation to 20 Hz, actual time intervals retained
Synthetic GNSS outages	10, 30, and 60 s; masks fixed before evaluation
Neural variants	Adaptive FusionNav (IMU+GNSS) and parameter-matched adaptive GNSS-only
Training seeds	0, 1, and 2
Deterministic baselines	Held-GNSS, constant velocity (CV), damped CV ($\tau = 5$ s selected on training episodes only)
Primary metric	Relative-motion RMSE _{3D} strictly inside each outage
Secondary metrics	Raw absolute 3D/horizontal/vertical RMSE, final outage error, recovery error

selected INSANE export, are aggregated to a 20 Hz recurrent-token timeline by time-weighted integration of past-held measurements rather than by simply discarding rows. Actual elapsed time, IMU sample age, and sample count are retained as context. The published timestamp corrections are treated as part of the dataset and are not applied a second time.

Ordinary GNSS is converted to a common local coordinate representation based on the dataset's geodetic information. The same first available ordinary-GNSS origin is used consistently per flight. No fitted transform based on the evaluation trajectory is applied. Before an artificial outage is generated, the complete final availability mask is fixed. All GNSS-derived quantities—position displacement, causal velocity, covariance/quality features, availability, and fix age—are then constructed using only the causally available source history. This ordering prevents hidden future satellite measurements from leaking into derivative features.

The causal velocity prior $\mathbf{v}^{CV}$ is estimated from the most recent 5 s of permitted GNSS fixes. The IMU and GNSS-only neural variants receive the same causally available GNSS history, timing, quality, and outage context. Scaling statistics are estimated from training flights only and then frozen. No ground-truth-derived transform, attitude, or reset is used in the model input or recurrent state.

3.3 Problem formulation and information constraints

Let t_a denote the time of the last causally available ordinary-GNSS source before an outage, $\mathbf{p}_a^G$ its position, and $\mathcal{J}_{0:t}$ the inertial history that is causally available up to time t . The navigation problem studied here is not unrestricted absolute localization. It is the conditional prediction of the vehicle's three-dimensional displacement after t_a using $\mathcal{J}_{0:t}$, the causally available

GNSS history $\mathcal{G}_{0:t_a}$, and non-positional context $\mathbf{c}_{0:t}$ such as measurement age and actual sampling intervals. During the blackout, no GNSS measurement whose source time is later than t_a may enter the predictor, either directly or through a derived velocity, covariance, or quality feature. The learned function can therefore be written abstractly as

$$\hat{\mathbf{d}}_t = F_{\theta}(\mathcal{J}_{0:t}, \mathcal{G}_{0:t_a}, \mathbf{c}_{0:t}), \quad t > t_a. \quad (1)$$

The corresponding absolute position estimate is $\hat{\mathbf{p}}_t = \mathbf{p}_a^G + \hat{\mathbf{d}}_t$ until satellite measurements become available again. This formulation makes the information boundary explicit and prevents an outage experiment from accidentally becoming a smoothing problem that uses postoutage information.

The matched ablation uses the same function class and exactly the same $\mathcal{J}_{0:t_a}$ and $\mathbf{c}_{0:t}$ but replaces only the six IMU values with zeros after the same train-only normalization. Consequently, the primary sensor-contribution quantity is not the standalone error of the full network, but the paired difference

$$\Delta E = E_{\text{full}} - E_{\text{GNSS-only}}. \quad (2)$$

A negative ΔE means that access to inertial measurements reduces error within the matched learned estimator. This definition avoids conflating the IMU contribution with benefits that can be obtained solely from causally available GNSS history, elapsed time, measurement age, recurrent memory, or network capacity.

The full model is intentionally not given reference attitude or inertial-navigation states. Therefore its IMU branch should be interpreted as a learned motion cue rather than as a replacement for calibrated strapdown mechanization. This distinction matters when comparing with classical INS: a physical inertial navigation filter estimates attitude, velocity, position, and sensor biases under a calibrated state model, whereas Adaptive FusionNav learns a direct residual motion representation under the input contract defined above.

3.4 Reproducibility controls

Several controls are used to keep the experiment auditable. Dataset archives, preprocessing code, configuration files, split definitions, outage masks, scalars, model sources, initial weight hashes, selected checkpoints, and output arrays are stored with hashes. Each outage prediction stores native reference timestamps, source GNSS timestamps, availability masks, and the exact reference rows used for scoring. The deterministic baselines are evaluated on the same timestamps and masks as the learned variants. Metric values in the result tables are independently recomputed from the saved prediction files after each Kaggle run.

The final test is additionally protected by a frozen-evaluation lock. The six adaptive checkpoints are copied unchanged from the completed validation experiment, their hashes are verified before and after evaluation, and the Damped-CV parameter is copied from its training-only selection record. An initial execution stopped before inference at an integrity check; the retry altered only the floating-point tolerance of a diagnostic manifest field. Scenario identifiers, timestamps, mask hashes, source hashes, checkpoints, metric definitions, and evaluation code remained unchanged, and no test metric was available before the repair.

These controls do not make the experiment statistically large, but they reduce a different risk: accidental disagreement between the code that generated the reported tables and the protocol described in the manuscript. In particular, they allow every abstract-level numerical claim in this paper to be traced to a saved held-out evaluation rather than to a manually selected notebook output.

3.5 Adaptive FusionNav architecture

Figure 1 summarizes the proposed adaptive FusionNav model. The design is intentionally compact. A six-dimensional IMU vector containing normalized accelerometer and gyroscope channels is projected to a 16-dimensional latent representation with a linear layer and hyperbolic tangent activation. A six-dimensional GNSS motion vector containing causal displacement/velocity information is projected through a second $6 \rightarrow 16$ branch. Ten shared context features, including availability, age of the last available fix, actual time interval, IMU timing/count information, and GNSS quality information, are projected through a $10 \rightarrow 8$ branch. The three representations are concatenated to a 40-dimensional vector and processed by a one-layer causal GRU with hidden dimension 32.

Two heads operate on the recurrent state $\mathbf{h}_t$. The first predicts a three-dimensional residual velocity,

$$\Delta \mathbf{v}_t = f_v(\mathbf{h}_t), \quad (3)$$

while the second predicts a scalar adaptive gate,

$$g_t = \sigma(f_g(\mathbf{h}_t)), \quad 0 \leq g_t \leq 1, \quad (4)$$

where $\sigma(\cdot)$ is the logistic sigmoid. The velocity used during the outage is

$$\hat{\mathbf{v}}_t = g_t (\mathbf{v}^{CV} + \Delta \mathbf{v}_t). \quad (5)$$

The predicted relative displacement is obtained by causal integration with the actual time interval,

$$\hat{\mathbf{d}}_{k+1} = \hat{\mathbf{d}}_k + \hat{\mathbf{v}}_k \Delta t_k. \quad (6)$$

At the first unavailable token, the elapsed interval includes the lag from the last available GNSS fix. When g_t approaches zero, predicted displacement is held rather than reset; when g_t approaches one, the decoder behaves like the ungated residual-velocity model. After GNSS becomes available again, the position decoder returns to the newly available satellite position while the recurrent hidden state remains continuous.

The model contains 7548 trainable parameters. The parameter-matched IMU-ablated comparator (hereafter GNSS-only for brevity) uses the same graph, the same number of parameters, and the same shared context. Only the six normalized IMU sensor values are zeroed. Thus, differences between the two variants are not caused by removing time, availability, quality, or other shared inputs. This matched construction is the primary mechanism used to isolate the incremental predictive value of the IMU within the learned architecture.

The gate should not be interpreted as a proven stop detector or as a physically identified mode switch. It is a learned attenuation coefficient trained jointly with the residual head. Its behavior is analyzed descriptively in the Results section, but no causal interpretation is assigned to individual gate values.

3.6 Outage construction and training augmentation

Training uses synthetic complete ordinary-GNSS outages inserted only into the training flights. Outage lengths are drawn from the predefined 10/30/60 s set with sufficient causally available history before the blackout and recovery time after it. The full and GNSS-only model of a given seed see the same episode order and masks, and their initial tensor hashes are matched. Across the adaptive experiment there are 540 frozen training episodes over at most 30 epochs.

Validation and test outages are not randomly regenerated after model outputs are observed. The validation set contains fixed 10, 30, and 60 s episodes on mars_4 and mars_5; the held-out test contains one fixed episode of each duration on each of mars_6 and mars_7. Durations on a single onset are related and should not be treated as independent flights. All methods are evaluated on identical native ground-truth rows for a given scenario.

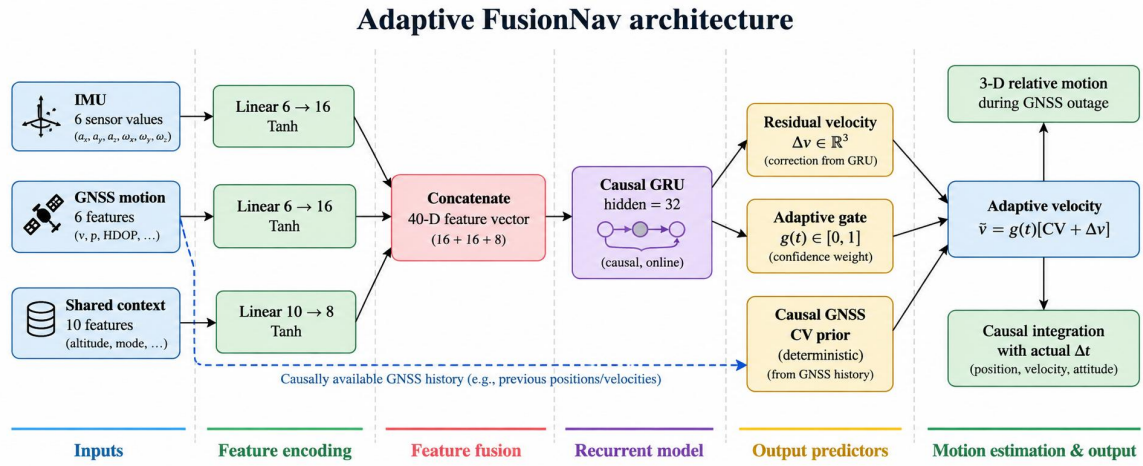


Fig. 1: Adaptive FusionNav architecture. The full model uses IMU, GNSS motion, and shared context. The matched GNSS-only model preserves the same architecture and parameter count but zeros only the six IMU sensor values. The gate $g(t)$ attenuates the combined causal velocity prior and learned residual before time integration.

3.7 Training and checkpoint selection

Six models are trained from scratch in the adaptive experiment: full and GNSS-only variants for seeds 0, 1, and 2. Optimization uses AdamW with learning rate 10^{-3} , weight decay 10^{-4} , batch size 4, gradient-norm clipping at 1, and Huber loss with transition parameter 1 m. The maximum training length is 30 epochs, and early stopping uses patience 5. For a paired full/GNSS-only seed, the training order and augmentation are matched until the models independently stop.

The checkpoint criterion is fixed before training: the macro arithmetic mean of relative-motion RMSE_{3D} across all fixed validation outage episodes. Strict improvement with zero minimum delta is required. The selected best epochs for the full model are 26, 18, and 30 for seeds 0, 1, and 2, respectively; the three GNSS-only models select epoch 18. These validation-selected checkpoints are frozen before test evaluation. There is no retraining on validation data and no checkpoint selection using test results.

3.8 Deterministic baselines

Three simple baselines are used because they expose whether the learned model improves on transparent kinematic assumptions.

Held-GNSS keeps the last available satellite position unchanged throughout the outage. It is a strong baseline when the vehicle moves little during the blackout.

Constant velocity estimates $\mathbf{v}^{CV}$ from the causally available GNSS history in the 5 s preceding the outage and propagates position linearly using actual elapsed time.

Damped constant velocity exponentially attenuates that prior,

$$\mathbf{v}(a) = \mathbf{v}^{CV} \exp(-a/\tau), \quad (7)$$

where a is outage age. Its analytical displacement is $\mathbf{v}^{CV} \tau (1 - \exp(-a/\tau))$. The damping constant is not selected on validation or test. A fixed candidate set $\{1, 2, 5, 10, 20, 30, 60, 120\}$ s is evaluated on 540 training episodes, and $\tau = 5$ s is frozen because it gives the smallest training macro relative-motion RMSE.

A classical error-state Kalman filter was implemented and passed synthetic mechanization checks during project development, but a defensible real-data configuration was not available because the supplied ordinary-GNSS-to-PX4-IMU lever arm and an independently validated absolute initial yaw were not available under the permitted sensor-input contract. Consequently, no numeric ESKF result is reported, and no claim of superiority over a calibrated INS/ESKF is made.

3.9 Evaluation metrics

The primary metric is designed to measure motion reconstruction during a complete GNSS outage while reducing sensitivity to the absolute error of the last available GNSS anchor. Let $\mathbf{p}_t$ denote the predicted absolute position, $\mathbf{p}_a^G$ the last available ordinary-GNSS anchor, $\mathbf{p}_t^R$ the reference position, and $\mathbf{p}_a^R$ the reference at the last available source time. The relative-motion error is

$$\mathbf{e}_t = (\mathbf{p}_t - \mathbf{p}_a^G) - (\mathbf{p}_t^R - \mathbf{p}_a^R). \quad (8)$$

For the set $\mathcal{O}$ of original reference rows strictly inside the blackout, the primary score is

$$\text{RMSE}_{3D} = \sqrt{\frac{1}{|\mathcal{O}|} \sum_{t \in \mathcal{O}} \|\mathbf{e}_t\|_2^2}. \quad (9)$$

Only the anchor reference may be interpolated to the last available source time; interpolated trajectory points are not counted as independent evaluation observations. No rigid or similarity alignment is performed on the predicted outage trajectory.

Secondary metrics include raw absolute 3D RMSE, horizontal and vertical RMSE, the error at the last unavailable point, the first 5 s recovery error, and the first error after an actual GNSS fix has returned. For neural methods, per-episode RMSE is first aggregated within each seed; the paper then reports the mean and sample standard deviation of the three seed-level values. The standard deviation is a repeat-training variability measure, not a confidence interval, and three seeds do not represent three independent flights.

4 Results

4.1 Validation performance

Table 2 reports the locked validation results after adaptive training. The primary controlled comparison is between Adaptive FusionNav and the parameter-matched IMU-ablated GNSS-only model. Across all validation episodes, the full model has lower error in 24 of 27 episode–seed comparisons, and each individual seed wins 8 of 9 fixed outage scenarios. Averaged by duration, the full model reduces the primary error from 6.340 to 5.147 m for 10 s outages, from 12.735 to 10.196 m for 30 s outages, and from 24.694 to 21.849 m for 60 s outages. These correspond to relative reductions of 18.8%, 19.9%, and 11.5%, respectively.

The validation comparison against simple baselines is heterogeneous by flight. On *mars_4*, Adaptive FusionNav improves on Held-GNSS, CV, and Damped-CV in average RMSE for all three outage durations. On *mars_5*, however, Held-GNSS and Damped-CV are lower at 10, 30, and 60 s. This difference is scientifically important: the adaptive network can model motion better on some flight regimes, but a stationary or heavily damped kinematic assumption is difficult to beat when the vehicle’s net displacement during the selected blackout is small. The validation results therefore support the matched IMU contribution but do not support a universal ranking over the deterministic baselines.

4.2 Held-out test performance

After all adaptive checkpoints, scalars, masks, metrics, and $\tau = 5$ s damping were frozen, the models were

evaluated on previously unused *mars_6* and *mars_7*. No learning or tuning occurred during this final evaluation. Table 3 contains the aggregate test results.

The held-out test independently confirms the key matched-ablation observation from validation. Enabling IMU measurements reduces the full-model RMSE by 34.0% at 10 s, 9.4% at 30 s, and 10.9% at 60 s relative to the matched GNSS-only neural model. Across six fixed test episodes and three seeds, the full model is better in 17 of 18 episode–seed pairs. Five of the six episodes are won by the full model for all three seeds. This is a materially stronger result than a single-seed improvement because the same directional effect persists after retraining and on flights that were not used for checkpoint selection.

The deterministic baselines provide an important complementary reference: on the aggregate held-out test, Held-GNSS, CV, and Damped-CV retain lower RMSE than Adaptive FusionNav at 10, 30, and 60 s. Thus, the principal empirical contribution of the learned model is the repeatable gain obtained from adding inertial information to a parameter-matched neural predictor, rather than a universal ranking over all motion priors. The baseline behavior is consistent with the selected test motion: holding is highly competitive when net displacement is small, while CV and Damped-CV remain strong over locally regular trajectories.

4.3 Per-flight test behavior

Table 4 separates the sensor-ablation effect from the overall method ranking. The neural IMU contribution is present on both test flights, but its magnitude varies. On *mars_6*, the largest mean full-versus-GNSS-only reduction occurs at 60 s (2.184 m), while at 10 s the reduction is 1.204 m. On *mars_7*, the 10 s episode is the single test case in which one seed does not improve with IMU; nevertheless, the mean full error remains 0.193 m lower than the GNSS-only mean. The 30 and 60 s reductions on *mars_7* are 1.232 and 2.021 m, respectively.

Figure 2 illustrates one representative held-out 60 s episode on *mars_6*. The plot is not selected as the “best” test result; it is the sole 60 s test outage on that flight. It illustrates why aggregate RMSE and simple baselines must accompany trajectory plots: different predictors can have similar average error while making qualitatively different directional mistakes. Figure 3 reports the pointwise relative-motion error for the same interval; neural curves are averaged across the three frozen seeds for visualization, whereas deterministic baselines are single traces. Together, the two figures show trajectory geometry and error growth without replacing the complete episode-level statistics.

Table 2: Validation relative-motion $RMSE_{3D}$ in metres. Neural entries are mean $\pm$ sample SD across three seed-level means; deterministic baselines have no seed SD. Lower is better.

Outage	Adaptive full	Adaptive GNSS-only	Held-GNSS	CV	Damped-CV
10 s	5.147 $\pm$ 0.104	6.340 $\pm$ 0.039	6.550	5.737	5.524
30 s	10.196 $\pm$ 0.479	12.735 $\pm$ 0.029	13.068	15.574	12.253
60 s	21.849 $\pm$ 1.730	24.694 $\pm$ 0.137	27.847	38.335	27.675

Table 3: Held-out test relative-motion $RMSE_{3D}$ in metres on mars_6 and mars_7. Neural entries are mean $\pm$ sample SD across the three frozen seeds.

Outage	Adaptive full	Adaptive GNSS-only	Held-GNSS	CV	Damped-CV
10 s	1.356 $\pm$ 0.097	2.055 $\pm$ 0.048	0.142	0.706	0.392
30 s	8.972 $\pm$ 0.153	9.908 $\pm$ 0.115	6.262	6.247	6.219
60 s	17.222 $\pm$ 0.925	19.325 $\pm$ 0.197	14.215	14.393	14.161

Table 4: Matched IMU contribution on validation and held-out test. Positive percentage denotes the relative reduction in RMSE of Adaptive FusionNav compared with the parameter-matched GNSS-only model. Win counts refer to episode-seed pairs.

Split	10 s improvement	30 s improvement	60 s improvement	Full wins
Validation	18.8%	19.9%	11.5%	24/27
Held-out test	34.0%	9.4%	10.9%	17/18

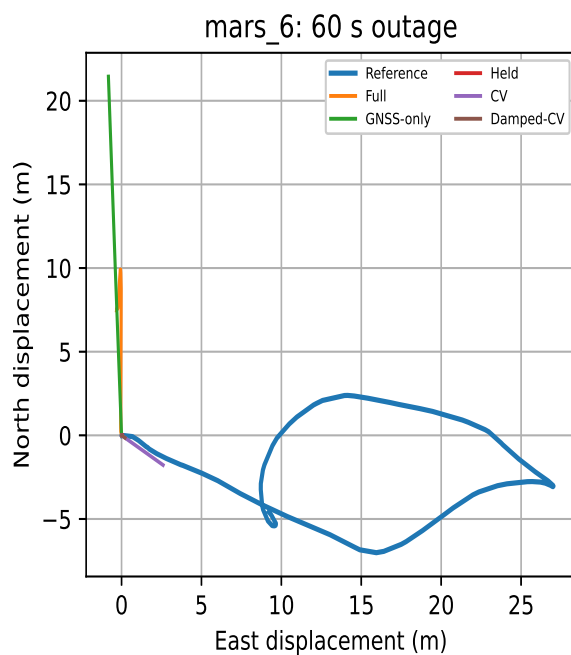


Fig. 2: Horizontal relative-motion trajectories for the held-out mars_6 60 s GNSS outage. The neural traces show seed 1 for visualization; all numerical conclusions use all three seeds.

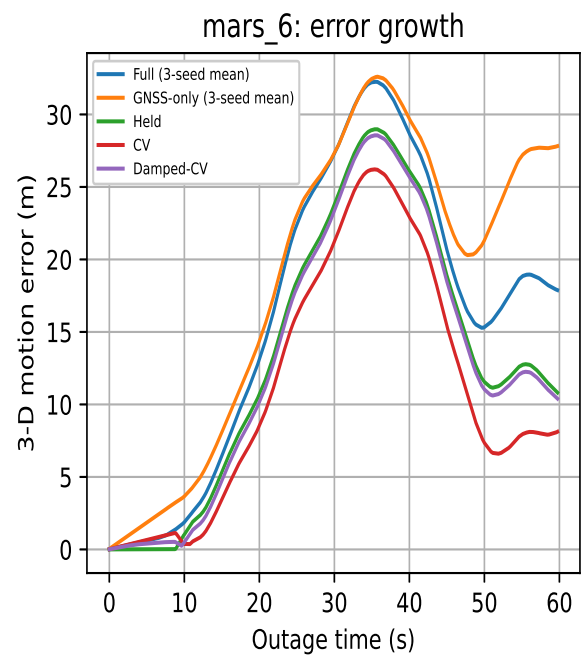


Fig. 3: Pointwise relative-motion error during the held-out mars_6 60 s outage. Neural curves are means of the three frozen seeds.

4.4 Adaptive gate behavior

The learned gate provides a descriptive view of how the network attenuates its velocity prediction as the satellite blackout progresses. Figure 4 displays the gate on the mars_6 60 s test episode for seed 1. The full and GNSS-only variants learn different attenuation profiles even though their architecture and parameter count are matched. Across the validation experiment, average gate values inside outages are generally larger for the full model than for the GNSS-only model, consistent with the full network retaining more motion when the IMU indicates continued dynamics. However, this interpretation is descriptive. Because the gate and residual-velocity head are jointly trained and no three-seed fixed-g ablation was run, the experiment does not isolate the gate as the causal source of improvement.

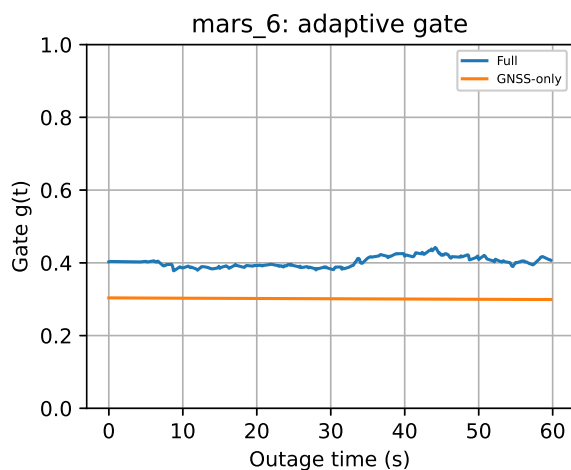


Fig. 4: Learned velocity attenuation gate for Adaptive FusionNav and the matched GNSS-only model in the held-out mars_6 60 s outage (seed 1). Gate values are descriptive model outputs, not a physically calibrated motion-state probability.

The computational footprint is small: each adaptive model has 7548 parameters and was evaluated on a Tesla T4 in the frozen environment. The saved timing values correspond to individual synchronized forward/readout measurements rather than a repeated hardware-latency benchmark; accordingly, we do not claim a statistically characterized onboard runtime. The compact parameter count and causal recurrent design nonetheless support future embedded evaluation without implying that such deployment has already been demonstrated.

5 Discussion

5.1 What the matched comparison establishes

The central experimental result is the consistency of the matched sensor ablation across model retraining and held-out flights. In the validation experiment, Adaptive FusionNav is lower than its GNSS-only counterpart in 24 of 27 episode-seed comparisons. More importantly, after the architecture and checkpoints are frozen, the same direction transfers to the held-out flights: 17 of 18 test episode-seed comparisons favor the IMU-enabled model. The relative reductions are largest for the 10 s aggregate test and remain present at 30 and 60 s. Because the two learned variants have identical parameter counts and preserve the same GNSS, timing, availability, quality, optimization, and augmentation context, the comparison provides evidence that the inertial values carry additional predictive information that the recurrent model can exploit.

The result establishes repeatable predictive value from the IMU within the controlled matched architecture. It does not by itself identify the adaptive gate as the sole cause of the gain or imply that the network reproduces calibrated strapdown mechanization; those questions require dedicated ablations and physical-state evaluation.

The distinction matters because the preceding model-development process revealed how easily an apparent sensor-fusion benefit can disappear when a GNSS-only control is added. A network with a position residual may learn static or trajectory-dependent GNSS-to-reference corrections without using the IMU in a meaningful way. The final model mitigates this risk by predicting residual velocity relative to a causal GNSS-motion prior and by using a matched ablation. The held-out test then provides an additional layer of protection against interpreting validation-specific patterns as general evidence.

5.2 Complementarity with simple kinematic baselines

A complementary result is that Held-GNSS, CV, and Damped-CV have lower aggregate test RMSE than Adaptive FusionNav at all three outage durations. This does not contradict the matched IMU result. The two questions are different. The full-versus-GNSS-only comparison asks whether inertial measurements improve a fixed learned architecture. The baseline comparison asks whether that architecture is the most accurate prediction rule for the selected test motion. The answer to the first is consistently positive in this experiment; the answer to the second is negative.

The baseline result is consistent with the motion in the held-out episodes. Low-displacement intervals favor Held-GNSS, while locally smooth trajectories make

constant-velocity propagation highly competitive. The relative-motion metric also removes constant anchor error by design, so transparent short-horizon motion priors form a particularly strong reference for this task.

The Damped-CV baseline is especially important because it guards against a misleading neural advantage created merely by reducing predicted speed as outage age increases. Its only parameter, $\tau = 5$ s, is selected exclusively on training episodes. The adaptive network still demonstrates a matched IMU effect, but on the held-out averages the damped baseline remains lower. This result argues against framing the learned gate as a novel replacement for physical navigation. A more accurate description is that it is a compact learned mechanism for combining a causal GNSS-motion prior with inertial context.

5.3 Relation to previous learning-assisted GNSS/INS work

Prior LSTM, GRU, CNN-RNN, transformer, and regression approaches have reported large improvements over pure INS or filter baselines under GNSS outages [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. Direct numeric ranking against those papers would be inappropriate because their platforms, sensors, target variables, outage lengths, and evaluation metrics differ. Many use land vehicles and predict pseudo-GNSS updates or INS errors inside a filter. Our architecture instead predicts an adaptively attenuated relative velocity and is evaluated on UAV flights with a matched GNSS-only ablation.

The primary methodological contribution relative to this literature is thus the experimental control rather than a claim that GRU cells or a sigmoid gate are themselves novel. Recurrent units are well established in navigation and sequence modeling. The new value of the present study is the combination of a compact causal fusion structure, a learned velocity attenuation mechanism, leakage-controlled outage construction, parameter-matched sensor ablation, flight-level separation, and one-shot held-out testing. This formulation makes it possible to state precisely what the experiment supports: inertial data improve the learned predictor within the matched architecture, even though the predictor is not the best method overall.

5.4 Scope, reference assumptions, and generalization

The experiment is intentionally scoped to the sensor contract available in INSANE. The dataset provides rich calibration material and a high-quality outdoor reference trajectory [31], but the ordinary-GNSS-to-PX4-IMU lever arm and an independently validated absolute initial yaw were not available in a form that supported a fully

calibrated real-data ESKF under the same input restrictions. We therefore omit a numerical ESKF comparison rather than report an under-specified filter. A calibrated INS/ESKF or factor-graph baseline with documented extrinsics and admissible heading initialization is an important next comparison on a fresh protocol.

The reference trajectory is derived by the dataset authors from dual RTK-GNSS and magnetometer geometry and is used only for targets and evaluation. To avoid treating interpolation as additional evidence, scoring uses original reference rows, with interpolation limited to the anchor time required by the relative-motion metric. Three random seeds quantify optimization variability but not mission-level uncertainty; the independent units for generalization are the flight sequences. Accordingly, the held-out mars_6/mars_7 test is the main confirmatory evaluation, while broader deployment generalization will require additional flights and datasets.

Finally, the present outages are controlled complete blackouts. Real degraded GNSS can include intermittent fixes, multipath, non-line-of-sight bias, jamming, spoofing, and time-varying covariance. Extending the same matched-ablation protocol to natural or hardware-in-the-loop degradations is therefore a direct next step.

5.5 Implications for intelligent 3D drone navigation

The results suggest a practical design lesson for intelligent UAV navigation. Learned fusion should not automatically replace transparent kinematic or physics-based estimators. Instead, it can be treated as an additional motion-prediction component whose value is validated against simple alternatives and isolated through sensor ablation. In the present experiment, the adaptive neural predictor makes demonstrably better use of IMU information than the matched GNSS-only model, but a robust system designer may prefer Held-GNSS or a kinematic prior when those rules are more accurate for a specific motion regime. Beyond trajectory estimation, Seitbattalov et al. considered runway-surface information in intelligent aircraft-landing decision support [46].

This observation motivates future hybrid selection or uncertainty-aware fusion rather than unconditionally using a single neural predictor. Such a future system might estimate when the learned IMU correction is reliable and when a simpler prior should dominate. Importantly, that idea is a future direction, not a result claimed by the current paper; adding a learned selector after observing the test would require new development data and a new independent evaluation.

6 Conclusion

This paper investigated intelligent 3D UAV navigation based on inertial and satellite data under controlled GNSS outages. Adaptive FusionNav combines a causal GNSS-motion prior, IMU measurements, shared timing/quality context, a compact GRU, a residual-velocity head, and a bounded adaptive gate. The model was evaluated on real INSANE UAV flights with complete flight-level separation between training, validation, and held-out testing and with a parameter-matched GNSS-only neural ablation.

The central result is a reproducible and held-out-tested contribution from inertial data within the matched neural fusion architecture. On the held-out test, the IMU-enabled model reduced relative-motion RMSE from 2.055 to 1.356 m at 10 s, from 9.908 to 8.972 m at 30 s, and from 19.325 to 17.222 m at 60 s, corresponding to improvements of 34.0%, 9.4%, and 10.9%. It achieved lower error than the matched GNSS-only model in 17 of 18 episode-seed comparisons. These test results are consistent with the matched validation trend.

The same experiments also show that the proposed model does not universally outperform simple navigation rules: Held-GNSS, constant velocity, and damped constant velocity remain lower on the aggregate held-out test. Accordingly, the scientific conclusion is not that Adaptive FusionNav is a universal replacement for classical navigation. Rather, the study demonstrates that inertial measurements provide repeatable additional predictive value inside a controlled causal neural fusion architecture. Future work should evaluate this effect on additional independent UAV datasets, include a fully calibrated real-data INS/ESKF baseline, and investigate uncertainty-aware switching between learned and transparent motion models without tuning to the present held-out test.

References

- [1] I. Jarraya, A. Al-Batati, M. B. Kadri, M. Abdelkader, A. Ammar, W. Boulila, and A. Koubaa, "GNSS-denied unmanned aerial vehicle navigation: analyzing computational complexity, sensor fusion, and localization methodologies," *Satellite Navigation* **6**, 9 (2025). <https://doi.org/10.1186/s43020-025-00162-z>.
- [2] C. Jiang, X. Zhou, H. Chen, and T. Liu, "UAV Positioning Using GNSS: A Review of the Current Status," *Drones* **10**(2), 91 (2026). <https://doi.org/10.3390/drones10020091>.
- [3] K. Ioannidi, Ch. Christakis, S. Sautbekov, P. Frangos, and S. K. Atanov, "The Radiation Problem from a Vertical Hertzian Dipole Antenna above Flat and Lossy Ground: Novel Formulation in the Spectral Domain with Closed-Form Analytical Solution in the High Frequency Regime," *International Journal of Antennas and Propagation* **2014**, 989348, 1–9 (2014). <https://doi.org/10.1155/2014/989348>.
- [4] N. Boguspayev, D. Akhmedov, A. Raskaliyev, A. Kim, and A. Sukhenko, "A Comprehensive Review of GNSS/INS Integration Techniques for Land and Air Vehicle Applications," *Applied Sciences* **13**(8), 4819 (2023). <https://doi.org/10.3390/app13084819>.
- [5] Y. He, J. Li, and J. Liu, "Research on GNSS INS & GNSS/INS Integrated Navigation Method for Autonomous Vehicles: A Survey," *IEEE Access* **11**, 79033–79055 (2023). <https://doi.org/10.1109/ACCESS.2023.3299290>.
- [6] F. Pöpl, H. Neuner, G. Mandlbürger, and N. Pfeifer, "Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: A framework and review," *ISPRS Journal of Photogrammetry and Remote Sensing* **196**, 287–305 (2023). <https://doi.org/10.1016/j.isprsjprs.2022.12.022>.
- [7] A. E. Kyzrkanov, S. K. Atanov, and S. A. R. Aljawarneh, "Formation control and coordination of swarm robotic systems," in *The 7th International Conference on Engineering & MIS 2021 (ICEMIS'21)*, 1–11 (2021). <https://doi.org/10.1145/3492547.3492704>.
- [8] G. Zhang and L.-T. Hsu, "Intelligent GNSS/INS integrated navigation system for a commercial UAV flight control system," *Aerospace Science and Technology* **80**, 368–380 (2018). <https://doi.org/10.1016/j.ast.2018.07.026>.
- [9] N. Cohen and I. Klein, "Inertial Navigation Meets Deep Learning: A Survey of Current Trends and Future Directions," *Results in Engineering* **24**, 103565 (2024). <https://doi.org/10.1016/j.rineng.2024.103565>.
- [10] A. Mohanty and G. Gao, "A survey of machine learning techniques for improving Global Navigation Satellite Systems," *EURASIP Journal on Advances in Signal Processing* **2024**, 73 (2024). <https://doi.org/10.1186/s13634-024-01167-7>.
- [11] W. Fang, J. Jiang, S. Lu, Y. Gong, Y. Tao, Y. Tang, P. Yan, H. Luo, and J. Liu, "A LSTM Algorithm Estimating Pseudo Measurements for Aiding INS during GNSS Signal Outages," *Remote Sensing* **12**(2), 256 (2020). <https://doi.org/10.3390/rs12020256>.
- [12] Z. Zhi, D. Liu, and L. Liu, "A performance compensation method for GPS/INS integrated navigation system based on CNN-LSTM during GPS outages," *Measurement* **188**, 110516 (2022). <https://doi.org/10.1016/j.measurement.2021.110516>.
- [13] Y. Tang, J. Jiang, J. Liu, P. Yan, Y. Tao, and J. Liu, "A GRU and AKF-Based Hybrid Algorithm for Improving INS/GNSS Navigation Accuracy during GNSS Outage," *Remote Sensing* **14**(3), 752 (2022). <https://doi.org/10.3390/rs14030752>.
- [14] S. Zhao, Y. Zhou, and T. Huang, "A Novel Method for AI-Assisted INS/GNSS Navigation System Based on CNN-GRU and CKF during GNSS Outage," *Remote Sensing* **14**(18), 4494 (2022). <https://doi.org/10.3390/rs14184494>.
- [15] M. S. El Sabbagh, A. Maher, M. A. H. Abozied, and A. M. Kamel, "Promoting navigation system efficiency during GPS outage via cascaded neural networks: A novel AI based approach," *Mechatronics* **94**, 103026 (2023). <https://doi.org/10.1016/j.mechatronics.2023.103026>.
- [16] Z. Yu, J. Jiang, P. Yan, Y. Li, J. Wu, and D. Xie, "A lightweight odometry network for GNSS/INS integration during GNSS outages," *Applied Soft Computing* **151**, 111143 (2024). <https://doi.org/10.1016/j.asoc.2023.111143>.
- [17] X. Kuang and B. Yan, "A Novel FECAM-iTransformer Algorithm for Assisting INS/GNSS Navigation System

- during GNSS Outages,” *Applied Sciences* **14**(19), 8753 (2024). <https://doi.org/10.3390/app14198753>.
- [18] Y. Zhu, M. Zhang, Y. Yang, C. Ran, and L. Zhou, “Improved Gaussian process regression-based method to bridge GPS outages in INS/GPS integrated navigation systems,” *Measurement* **229**, 114432 (2024). <https://doi.org/10.1016/j.measurement.2024.114432>.
- [19] Y. Wang, Q. Wang, H. Zhang, and X. Duan, “A Low-Cost GNSS/INS integration method aided by Cascade-LSTM Pseudo-Velocity measurement for bridging GNSS outages,” *Measurement* **240**, 115518 (2025). <https://doi.org/10.1016/j.measurement.2024.115518>.
- [20] F. Liu, H. Zhao, and W. Chen, “A Hybrid Algorithm of LSTM and Factor Graph for Improving Combined GNSS/INS Positioning Accuracy during GNSS Interruptions,” *Sensors* **24**(17), 5605 (2024). <https://doi.org/10.3390/s24175605>.
- [21] F. Wu, H. Luo, F. Zhao, L. Wei, and B. Zhou, “Optimizing GNSS/INS Integrated Navigation: A Deep Learning Approach for Error Compensation,” *IEEE Signal Processing Letters* **31**, 3104–3108 (2024). <https://doi.org/10.1109/LSP.2024.3484292>.
- [22] C. Chen, X. Lu, A. Markham, and N. Trigoni, “IONet: Learning to Cure the Curse of Drift in Inertial Odometry,” in *Proc. AAAI Conference on Artificial Intelligence* **32**(1) (2018). <https://doi.org/10.1609/AAAI.V32I1.12102>.
- [23] H. Yan, Q. Shan, and Y. Furukawa, “RIDI: Robust IMU Double Integration,” in *Computer Vision – ECCV 2018*, 641–656 (2018). https://doi.org/10.1007/978-3-030-01261-8_38.
- [24] S. Herath, H. Yan, and Y. Furukawa, “RoNIN: Robust Neural Inertial Navigation in the Wild: Benchmark, Evaluations, and New Methods,” in *Proc. IEEE International Conference on Robotics and Automation*, 3146–3152 (2020). <https://doi.org/10.1109/ICRA40945.2020.9196860>.
- [25] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, “TLIO: Tight Learned Inertial Odometry,” *IEEE Robotics and Automation Letters* **5**(4), 5653–5660 (2020). <https://doi.org/10.1109/LRA.2020.3007421>.
- [26] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU Dead-Reckoning,” *IEEE Transactions on Intelligent Vehicles* **5**(4), 585–595 (2020). <https://doi.org/10.1109/TIV.2020.2980758>.
- [27] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe-based visual-inertial odometry using nonlinear optimization,” *International Journal of Robotics Research* **34**(3), 314–334 (2015). <https://doi.org/10.1177/0278364914554813>.
- [28] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On-Manifold Preintegration for Real-Time Visual-Inertial Odometry,” *IEEE Transactions on Robotics* **33**(1), 1–21 (2017). <https://doi.org/10.1109/TRO.2016.2597321>.
- [29] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator,” *IEEE Transactions on Robotics* **34**(4), 1004–1020 (2018). <https://doi.org/10.1109/TRO.2018.2853729>.
- [30] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem,” in *Proc. AAAI Conference on Artificial Intelligence* **31**(1) (2017). <https://doi.org/10.1609/AAAI.V31I1.12125>.
- [31] C. Brommer, A. Fornasier, M. Scheiber, J. Delaune, R. Brockers, J. Steinbrener, and S. Weiss, “The INSANE dataset: Large number of sensors for challenging UAV flights in Mars analog, outdoor, and out-/indoor transition scenarios,” *International Journal of Robotics Research* **43**(8), 1083–1113 (2024). <https://doi.org/10.1177/02783649241227245>.
- [32] A. Kyziryanov, S. Atanov, and S. Aljawarneh, “Coordination of movement of multiagent robotic systems,” in *Proc. 2021 16th International Conference on Electronics Computer and Computation (ICECCO)*, 1–4 (2021). <https://doi.org/10.1109/ICECCO53203.2021.9663796>.
- [33] A. Antonopoulos, M. G. Lagoudakis, and P. Partsinevelos, “A ROS Multi-Tier UAV Localization Module Based on GNSS, Inertial and Visual-Depth Data,” *Drones* **6**(6), 135 (2022). <https://doi.org/10.3390/drones6060135>.
- [34] Z. Qiu, J. Lv, D. Lin, Y. Yu, Z. Sun, and Z. Zheng, “A Lidar-Inertial Navigation System for UAVs in GNSS-Denied Environment with Spatial Grid Structures,” *Applied Sciences* **13**(1), 414 (2023). <https://doi.org/10.3390/app13010414>.
- [35] A. K. Kereyev, S. K. Atanov, K. P. Aman, Z. K. Kulmagambetova, and B. T. Kulzhagarova, “Navigation system based on Bluetooth beacons: Implementation and experimental estimation,” *Journal of Theoretical and Applied Information Technology* **98**(8), 1187–1200 (2020). <https://www.jatit.org/volumes/Vol98No8/6Vol98No8.pdf>.
- [36] G. Zhao, Y. Wang, and X. Wang, “Error compensation method of GNSS/INS integrated navigation system based on PSO-LSTM,” *Advances in Space Research* **75**(12), 8657–8666 (2025). <https://doi.org/10.1016/j.asr.2025.04.024>.
- [37] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, “Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review,” *Sensors* **20**(15), 4220 (2020). <https://doi.org/10.3390/s20154220>.
- [38] S. S. Brimzhanova, S. K. Atanov, Kh. Moldamurat, K. S. Kobelev, and L. G. Gagarina, “Cross-platform compilation of programming language Golang for Raspberry Pi,” in *Proc. 5th International Conference on Engineering and MIS (ICEMIS’19)*, article 10, 1–5 (2019). <https://doi.org/10.1145/3330431.3330441>.
- [39] A. G. Serek, A. Issabek, A. Akhmetov, and A. Sattarbek, “Part-of-speech tagging of Kazakh text via LSTM network with a bidirectional modifier,” in *Proc. 2021 16th International Conference on Electronics Computer and Computation (ICECCO)*, 1–4 (2021). <https://doi.org/10.1109/ICECCO53203.2021.9663794>.
- [40] S. Akhmetzhanova, A. Serek, R. Kashayev, and A. Kozhamuratova, “Few-shot brain tumor classification: meta-vs metric-learning comparison,” *Bulletin of Electrical Engineering and Informatics* **14**(5), 3913–3922 (2025). <https://doi.org/10.11591/eei.v14i5.10706>.
- [41] L. Zholshiyeva, T. Zhukabayeva, A. Serek, R. Duisenbek, M. Berdieva, and N. Shapay, “Deep Learning-Based Continuous Sign Language Recognition,” *Journal of Robotics and Control* **6**(3), 1106–1119 (2025). <https://doi.org/10.18196/jrc.v6i3.25881>.
- [42] A. Serek and M. Zhaparov, “Optimizing preference satisfaction with genetic algorithm in matching students

- to supervisors,” *Applied Mathematics & Information Sciences* **18**(1), 133–138 (2024). <https://doi.org/10.18576/amis/180114>.
- [43] D. N. Kuanyshbay, A. G. Serek, A. A. Shoiynbek, K. R. Sharipov, T. A. Shoiynbek, B. A. Meraliyev, and M. A. Meraliyev, “Development of an AI-Based Communication Fraud Detection System,” *Applied Mathematics & Information Sciences* **19**(4), 953–963 (2025). <https://doi.org/10.18576/amis/190419>.
- [44] A. G. Serek, K. Orynbekova, A. Talasbek, D. Kariboz, G. Saimassay, and A. Bogdanchikov, “Recommendation System for Human Resource Management by the Use of Apache Spark Cluster,” in *Proc. 2023 17th International Conference on Electronics Computer and Computation (ICECCO)*, 1–4 (2023). <https://doi.org/10.1109/ICECCO58239.2023.10147129>.
- [45] A. G. Serek, M. K. Zhaparov, and S.-M. Yoo, “Analysis of data to improve system of an educational organization,” in *Proc. 2018 14th International Conference on Electronics Computer and Computation (ICECCO)*, 206–212 (2018). <https://doi.org/10.1109/ICECCO.2018.8634709>.
- [46] Z. Y. Seitbattalov, S. K. Atanov, and Z. S. Moldabayeva, “An Intelligent Decision Support System for Aircraft Landing Based on the Runway Surface,” in *Proc. 2021 IEEE International Conference on Smart Information Systems and Technologies (SIST)*, 1–5 (2021). <https://doi.org/10.1109/SIST50301.2021.9466000>.



Ayan Kemel received the Master of Engineering degree from L.N. Gumilyov Eurasian National University, Astana, Kazakhstan. He is currently a doctoral student in the Department of Computer and Software Engineering at L.N. Gumilyov Eurasian National University. His research

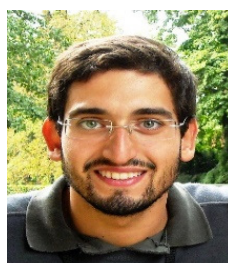
interests include intelligent information systems, embedded computing, automation, sensor fusion, and data-driven navigation. In the present study, he contributed to experimental implementation, software development, data preparation, model evaluation, visualization, and manuscript preparation. ORCID: 0009-0008-2074-0942. E-mail: kemel.am.1@enu.kz.



Sabyrzhan Atanov

is Doctor of Technical Sciences and Professor in the Department of Computer and Software Engineering at L.N. Gumilyov Eurasian National University, Astana, Kazakhstan. He has led nationally supported research projects on intelligent

systems, artificial intelligence, embedded systems, and microcontroller-based platforms. His interests include system modelling, automation, and computational intelligence for engineering applications. In the present study, he contributed to conceptual development, scientific supervision, methodological review, and interpretation of results. ORCID: 0000-0003-2115-7130. E-mails: atanov_sk@enu.kz; atanov5@mail.ru.



Luigi La Spada

is an Assistant Professor in the School of Computing, Engineering and the Built Environment at Edinburgh Napier University, United Kingdom. He received the PhD degree in Electrical and Electronic Engineering from the University of

Pennsylvania and Roma Tre University and an MSc in Blended and Online Education from Edinburgh Napier University. His interdisciplinary research links machine learning, autonomous systems, photonics, cybersecurity, and trustworthy cyber-physical systems. He has contributed to European and UK research projects and is active in IEEE, IET, and URSI. In the present study, he contributed to scientific review, interpretation of results, and manuscript revision. ORCID: 0000-0003-4923-1320. E-mail: L.LaSpada@napier.ac.uk.