

# A Comparative Analysis of Hybrid Sensor Fusion Schemes for Visual–Inertial Navigation

Tarafder Elmi Tabassum, Ivan Petrunin, and Zeeshan A. Rana

**Abstract**— Visual Inertial Odometry (VIO) has been extensively studied for navigation in GNSS-denied environments, but its performance can be heavily impacted by the complexity of the navigation environments such as weather conditions, illumination variation, flight dynamics, and environmental structure. Hybrid fusion approaches integrating Neural Networks (NN), especially Gated Recurrent units (GRU) with the Kalman filters (KF), such as Error-State Kalman Filter (ESKF) have shown promising results mitigating system nonlinearities due to challenging environmental conditions data issues, there is a lack of systematic studies quantitatively analysing and comparing performance differences unhand. To address this gap and enable robust navigation in complex conditions, this study proposes and systematically analyses the performance of three hybrid fusion schemes for VIO-based navigation of Unmanned Aerial Vehicles (UAV). These three hybrid VIO schemes include Visual Odometry (VO) error compensation using NN, KF error compensation using NN, and prediction of Kalman gain using NN. The comparative analysis is performed using data generated in MATLAB incorporating the Unreal Engine involving diverse challenging environmental conditions: fog, rain, illumination level variability and variability in the number of features available for extraction during the UAV flight in the urban environment. The results demonstrate the performance improvement achieved by hybrid VIO fusion schemes compared to ESKF-based traditional fusion methods in the presence of multiple visual failure modes. Comparative analysis reveals notable improvement achieved by method 1 with enhancements of 93% in sunny, 91% in foggy and 90% in rainy conditions than the other two hybrid VIO architectures.

**Index Terms**— Hybrid VIO, Neural Networks, Kalman Filter, Failure modes, Error compensation, Robustness.

## I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) are increasingly prevalent in Urban Air Mobility (UAM) applications including delivery drones, passenger taxis and medical goods delivery. As urban congestion escalates, the impending integration of UAM into smart cities facilitates the necessity of exploring navigation methods to optimize aerial transport potential. This exploration includes enabling autonomous navigation capabilities, which demand precise positioning for seamless operation, particularly when navigating between buildings in urban environments. Visual Inertial Navigation Systems (VINS) have gained significant interest among

researchers as a prominent alternative solution due to Navigation Satellite Systems (GNSS) vulnerability in urban environments. Current state-of-the-art includes various categories of Visual-Inertial Odometry (VIO) positioning approaches: feature-based VIO, data-driven VIO, direct VIO, and hybrid VIO methods. It was discovered that existing vision-based alternative solutions are highly sensitive to the complexity of navigation environments such as weather effects, illumination variation, trajectory dynamics and environment structure. While fusion techniques such as filter-based VIO and data-driven VIO have generated moderate results throughout their evaluation, they are nevertheless vulnerable when deployed in complex navigation environments characterized by challenging conditions like motion blur [1], [2], low texture [3], [4], illumination variation [5], [6], [7], [8] rapid motion [9], [10], [11] and weather conditions [12], [13], [14], [15]. However, researchers integrated camera with other sensors like lidar and radar for UAVs navigating in adverse weather conditions [16], [17], [18]. Another work on adverse weather conditions using the Convolutional Neural Network with autoencoders to detect haze levels and improve image quality for autonomous navigation yet suffers to provide promising results due to a lack of generalization ability [19]. Besides, modelling the effects can be challenging, and the lack of feature tracking led to divergence in the optimization process which data-driven fusion techniques may not eliminate.

Most of the studies like [20], [21] researchers have not focused on evaluating the filter-based VIO under complex conditions such as lighting variations and weather effects. One of the drawbacks of filter-based VIO such as variants of Kalman filter (KF) is that the linear assumptions model and Gaussian noise may not hold under dynamic conditions, leading to divergences in the state estimation. Moreover, classical VIO often requires parameter tuning, linear optimisation, and calibration for effective operation. In this work, the imperfect model, and system nonlinearity of the Error-State Kalman filter (ESKF) under complex conditions are investigated and performance enhancement has been highlighted through mitigating challenges by maintaining parameter linearization. The continued development of machine learning has enabled the direct processing of high-dimensional data and the learning of underlying models that can enhance state estimation. Nevertheless, the data-driven VIO methods are characterized by lower performance in scenarios that differ significantly from the training datasets [22], [23].

Performance advantage that can be achieved by integrating Neural Networks (NN) with KF in dynamic conditions has indicated the potential benefit of adopting hybrid approaches in

The authors are with the School of Aerospace, Transportation and Manufacturing, Cranfield University, Cranfield, Bedfordshire, United Kingdom, MK43 0AL (email: tarafder-elmi.tabassum@cranfield.ac.uk; i.petrinin@cranfield.ac.uk; zeeshan.rana@cranfield.ac.uk)

VIO systems [8]. Hybrid integrations enhance the ability to manage complex scenarios by evaluating the level of epistemic uncertainty, adapting to changing sensor conditions and identifying fault patterns. Strategies such as enhancing the model's architecture, refining its parameters, and expanding training datasets are being reviewed for machine learning-based VIO, with hybrid approaches offering potential solutions to minimize these requirements. Several hybrid (NN-KF) VIO methods have provided the advantages of using time-dependent recurrent NN models like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU) and Bi-directional LSTM (Bi-LSTM) in VIO as state-predictors [12], [24] and error compensators [25], [26], [27]. The authors of these studies have investigated VIO performance at different fault conditions arising while navigating complex environments, yet the generalization ability of such systems needs to be explored. Kim et. al [28] have reviewed hybrid NN-KF techniques in a broad range of applications and categorized them according to the role of NN applied to KF, as follows- 1) prediction of tuning parameters using NN, 2) error compensation using NN, and 3) update observation model of KF using NN as illustrated Fig. 1.

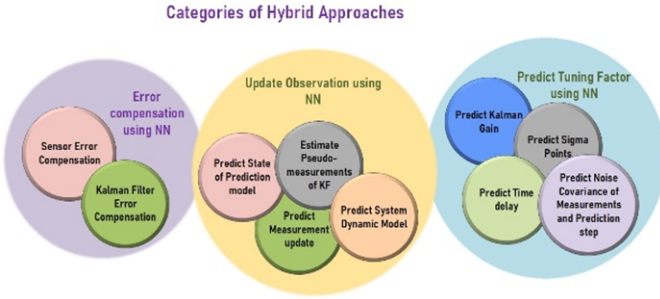


**Fig. 1.** Categories of hybrid NN-KF techniques according to [28]

Authors of [29], and [30] have outlined recommendations concerning improving accuracy, robustness, and generalization ability for the systems utilizing hybrid NN-KF approaches to mitigate sensor error, imperfections in system models, and nonlinearity in dynamic conditions. However, to the best of the author's knowledge, studies on quantitative differences between various types of hybrid architectures have not shown any indications of investigating performance variations and generalization ability in order to provide robust conclusions regarding the pros and cons of each architecture. Therefore, this work performs a comparative analysis of three types of hybrid NN-KF approaches implemented in VIO systems with a focus on evaluating their capability to mitigate multiple visual failure modes to ensure integrity while navigating complex urban environments. To support the comparative analysis, this paper also proposed a hybrid methodology in VIO application that predicts increment of Kalman filter to mitigate imperfect modelling and system nonlinearities. The proposed scheme integrates GRU as the KF error compensator at the output of ESKF-based VIO. Additionally, three different hybrid architectures for VIO fusion considered in this work are frequently used for similar applications according to findings of work [28] that include VO error compensation using NN [26], KF error compensation using NN [27] and prediction of

Kalman gain using NN [31]. However, the third method in the comparative analysis is utilized from the current state-of-the-art solution, KalmanNet [31] which has employed cascade-GRU for predicting the k-gain of EKF. In this research, the three different hybrid architectures are designed as a combination of GRU and ESKF where GRU is utilized for different purposes for each hybrid approach to enhance the efficiency of the Kalman filter. In addition, the formulation is extended by providing robustness analysis on these methods using MATLAB incorporated with Unreal Engine simulated photorealistic datasets replicating possible fault conditions present in urban environments including weather effects. The contributions of this work are highlighted as follows –

1. A novel hybrid VIO architecture is proposed based on ESKF and GRU-based error correction mechanism. To the best of the authors knowledge, the combination was not proposed in the literature for VIO applications. It has been identified as an effective approach for mitigating errors arising from predefined models.
2. A comparative study of the three proposed hybrid VIO schemes utilizing the same technologies, i.e. GRU and ESKF, but in different architectures to evaluate their performance in the same dataset across challenging environmental conditions affecting vision feature extraction, such as fog, rain, illumination issues, as well as texture-less conditions and motion effects.
3. Systematic evaluation of resilience of each hybrid VIO algorithms to common degradation modes including feature extraction errors, feature tracking errors, feature mismatch, feature location errors and data association errors.
4. Evaluation of performance under “out of distribution” problems for each algorithm aiming to estimate their ability to adapt and perform effectively in unseen scenarios, providing insights into their robustness and versatility.

The remainder of this paper is structured as follows: section 2 includes three hybrid visual-inertial navigation architectures with brief descriptions. Section 3 discusses experimental settings for data generation to train and test the hybrid models. Section 4 discusses the test and performance analysis of the selected algorithms under diverse complex scenarios with comparison. Finally, the conclusion is presented in section 5.

## II. HYBRID VISUAL-INERTIAL NAVIGATION ARCHITECTURES

This section provides a brief overview of the hybrid architectures implemented in VINS systems. In line with the research objective of assessing the strengths and limitations of the hybrid VIO solutions in complex environments including the influence of visual failure modes, three adopted hybrid VIO fusion architectures are as follows- 1) GRU-aided ESKF-VIO with VO error compensator (Method 1), 2) GRU-aided ESKF-VIO with KF error compensator (Method 2) and (3) GRU-aided ESKF-VIO with Kalman gain predictor (Method 3). These architectures are illustrated in Fig. 1.

The explored hybrid ESKF-based VIO fusion scheme is well-known for its ability to minimize linearization errors,

handle sensor delay, and efficiently model errors. The GRU is implemented to enhance the performance of ESKF, chosen due to its effectiveness in processing time-varying sequences compared to other NN. The GRU includes an update gate  $z_t$  that regulates the influence of the previous state on the current state and a reset gate that determines the extent to which past hidden information is forgotten. The GRU architecture is presented in Fig. 3. This work has utilized single-layer GRU architecture adopted from [32] where hyperparameter is tuned based on input datasets for each architecture. A fully connected layer is used to provide the output from the model.

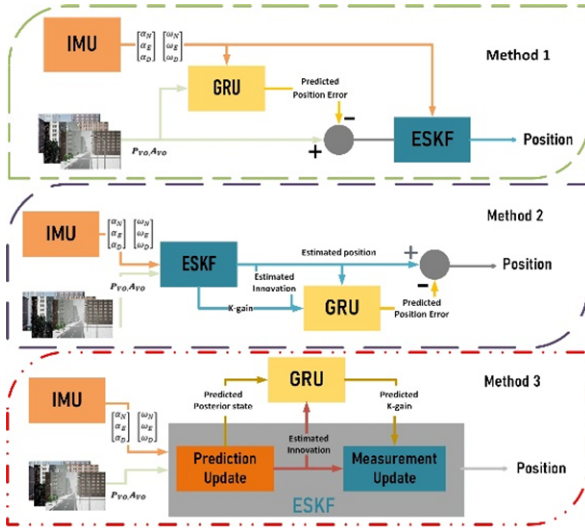


**Fig. 2.** Architecture of the hybrid VIO techniques utilised for comparison: (a) GRU-aided ESKF-VIO with VO error compensator as method 1, b) GRU-aided ESKF-VIO with KF error compensator as method 2, c) GRU-aided ESKF-VIO with Kalman gain predictor as method 3.

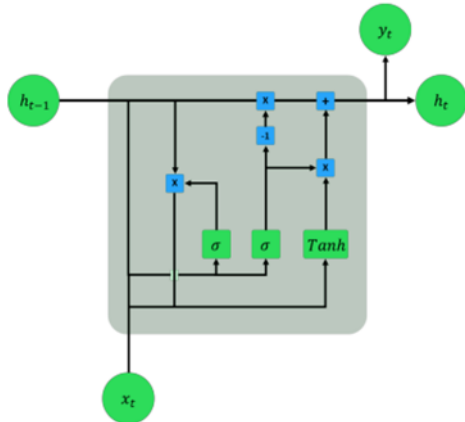


**Fig. 3.** Diagram for Gated Recurrent Unit (GRU)

The following subsections are organized as follows- section 2.A describes traditional ESKF fusion methodology which is utilized with a combination of GRU to formulate proposed Hybrid VIO architectures; section 2.B describes Method 1; section 2.C discusses proposed Method 2 and lastly, section 2.D describes the methodology of Method 3.

#### A. Error State Kalman Filter

The main concept of the ESKF fusion strategy consists of the

separation of the state vector into two parts: nominal state  $x$ , and error state  $\delta x$ .

$$x_k = x + \delta x \quad (1)$$

Notably, the frame of references used to develop navigation equations for state estimation of ESKF follows the north-east-down (NED) frame format.

The estimated state vector  $x$  is written as

$$x = [p \ v \ q \ \alpha_b \ \omega_b]^T \quad (2)$$

Where,  $p = [p_N, p_E, p_D]$ - represents the position in the platform;  $v = [v_N, v_E, v_D]$ - represents the platform's velocity;  $q = [q_N, q_E, q_D]$ - represents orientation in the form of quaternion;  $\alpha_b = [\alpha_{b_N}, \alpha_{b_E}, \alpha_{b_D}]$ - represents accelerometer drift and  $\omega_b = [\omega_{b_N}, \omega_{b_E}, \omega_{b_D}]$ - represents gyroscope drift accordingly.

The error state vector  $\delta x$  can be represented as

$$\delta x = [\delta p \ \delta v \ \delta q \ \delta \alpha_b \ \delta \omega_b]^T \quad (3)$$

Where,  $\delta p = [\delta p_N, \delta p_E, \delta p_D]$ - represents position error in the platform;  $\delta v = [\delta v_N, \delta v_E, \delta v_D]$ - represents the platform's velocity error;  $\delta q = [\delta q_N, \delta q_E, \delta q_D]$ - represents the angular error of the form of quaternion;  $\delta \alpha_b = [\delta \alpha_{b_N}, \delta \alpha_{b_E}, \delta \alpha_{b_D}]$ - represents accelerometer drift error and  $\delta \omega_b = [\delta \omega_{b_N}, \delta \omega_{b_E}, \delta \omega_{b_D}]$  represents gyroscope drift error accordingly. The conversation between quaternion and angular representations is adapted from the ESKF model explained in [33]. The discretized nominal state kinematics equations are as follows-

$$\begin{aligned} p_k &= p_{k-1} + v_{k-1}\Delta t + (R(q_{k-1})(a_m - a_{b_k}) + g) \frac{\Delta t^2}{2}; \\ v_k &= v_{k-1} + (R(q_{k-1})(a_m - a_{b_k}) + g)\Delta t; \\ q_k &= q_{k-1} \otimes q((\omega_m - \omega_{b_k})\Delta t); \\ a_{b_k} &= a_{b_{k-1}}; \\ \omega_{b_k} &= \omega_{b_{k-1}} \end{aligned} \quad (4)$$

And discretized error state kinematics equations are as follows-

$$\begin{aligned} \delta p &= \delta p_{k-1} + \delta v_{k-1}\Delta t; \\ \delta v_k &= \delta v_{k-1} + (-R\{q_{k-1}\}[a_m - a_{b_k}]_X \delta q_{k-1}) \cdot \Delta t; \\ \delta q_k &= R^T\{(\omega_m - \omega_{b_k})\Delta t\} \delta q_{k-1} - \delta \omega_{b_k} \Delta t; \\ \delta a_{b_k} &= \delta a_{b_{k-1}} + \alpha_i; \\ \delta \omega_{b_k} &= \delta \omega_{b_{k-1}} + \omega_i \end{aligned} \quad (5)$$

Where,  $k, k-1$ - indexes of current and previous sample step;  $g = [0 \ 0 \ -9.81]$ - represents gravity vector and  $R\{q\}$ - represents quaternion to rotation matrix conversion as defined in [33].

The Jacobians are obtained by simple inspection of error state differences equations as follows-

$$F_x = \left. \frac{\partial f}{\partial \delta x} \right|_x = \begin{bmatrix} I & I\Delta t & 0 & 0 & 0 \\ 0 & I & -R[a_m - a_{b_k}]_X \Delta t & -R\Delta t & 0 \\ 0 & 0 & R^T\{(\omega_m - \omega_{b_k})\Delta t\} & 0 & -I\Delta t \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & I \end{bmatrix} \quad (6)$$

The ESKF states as follows-

$$\delta x = F_x(x, u_m)\delta x + F_i i$$

where,  $F_i$  denotes noise states,  $i$  denotes noise vector and  $u_m$

denotes IMU measurements.

The error states and the covariance prediction process can be obtained as

$$\begin{aligned}\delta\hat{x} &= F_x(x, u_m)\delta\hat{x} \\ P &= F_x P_{k-1} F_x^T + F_i Q_i F_i^T\end{aligned}\quad (7)$$

Where,  $Q_i$  represents the noise covariance matrix-

$$Q_i = \begin{bmatrix} \sigma_a^2 \Delta t^2 I & 0 & 0 & 0 \\ 0 & \sigma_\omega^2 \Delta t^2 I & 0 & 0 \\ 0 & 0 & \sigma_{a_k}^2 \Delta t I & 0 \\ 0 & 0 & 0 & \sigma_{\omega_k}^2 \Delta t I \end{bmatrix}$$

The standard deviations for Gaussian noise for acceleration, angular velocity, acceleration drift and angular velocity drift are denoted by  $\sigma_a$ ,  $\sigma_\omega$ ,  $\sigma_{a_k}$ ,  $\sigma_{\omega_k}$  over the step time  $\Delta t$  respectively. Additionally,  $I$  represents the eye vector and  $T$  represents the transpose operation of the vector.

In this paper, each hybrid ESKF-aided fusion scheme integrates pose measurements generated by VO with linear acceleration from the accelerometer and angular velocity from the gyroscope. Featured-based VO is preferred for the research due to its simplicity and reliability, contrasting with data-driven VO which requires more computational resources and extensive training data. Once the VO estimated position becomes available, the observation is incorporated to continuously update the filter. The process involves calibrating accelerometer and gyroscope drifts. The observation is expressed as follows-

$$y_k = h(x_t) + v$$

Where,  $y_k$ -represents measurement vector,  $h()$ - represents system state and  $v$ -represents white Gaussian noise with Covariance  $R$ .

The measurement update equations are derived as

$$\begin{aligned}K &= \hat{P}_{k-1} H^T (H \hat{P}_{k-1} H^T + R)^{-1} \\ \delta\hat{x} &= K(y_k - h(\hat{x}_t)) \\ \hat{P} &= (I - KH)\hat{P}_{k-1}\end{aligned}\quad (8)$$

The Jacobian matrix  $H$  for filter correction consists of standard Jacobian of  $h()$  and true state Jacobian concerning the error state which is denoted as

$$H \triangleq \left. \frac{\partial h}{\partial x} \right|_x \left. \frac{\partial x_t}{\partial \delta x} \right|_x = H_x X_{\delta x}$$

The following sections elaborate on three distinct methods for integrating GRU within the ESKF model to enhance fusion capacities.

### B. Method 1

The concept of compensating sensor error using NN is utilized to enhance the performance of KF in this integration. The visual sensor has prominent fault factors contributing to performance degradations in traditional KF-fusion techniques. The previous work on the tightly coupled GRU-aided ESKF based VIO has utilized error compensation mechanism to corrected VO data. The integration was part of a fault-tolerant FMEA-supported federated multi-sensor navigation system. The federated multi-sensor fusion aims to enable uninterrupted operation of UAV under various fault conditions including illumination variation, motion variation, low-texture environment, and flight dynamics. The paper adopted hybrid fusion phenomena from the previous work. Such design could be beneficial to correct imperfect sensors, parameters or errors

of the sensor caused by its characteristics (VO drift for longer flights). The presence of outliers, feature tracking errors and illumination variation mostly occurred when weather changes and such a design could improve system robustness under complex scenarios. In this paper, the visual sensor with degraded performance is corrected using GRU predicted position increment incorporating following equation before ESKF fusion-

$$P_{corrected} = P_{VO} - \Delta P_{VO/GRU} \quad (9)$$

where,  $P_{VO/GRU}$  -represents corrected VO position,  $P_{VO}$  -represents position from VO and  $\Delta P_{VO/GRU}$  -represents GRU predicted VO position increments.

The GRU model is adapted from previous work consisting of a single layer with 400 neurons and the sequence length input. The model is trained with a batch of 64, a dropout rate of 0.5, and tanh activation function. The ADAMAX optimizer is employed for training while the lose function is calculated using mean square error (MSE).

### C. Method 2

This hybrid fusion architecture proposes compensating the ESKF errors using errors prediction generated by GRU. This fusion scheme is beneficial to correct model error estimated by KF by predicting unmodelled error using NN. Villalonga Torres et al. [27] have proposed Bi-LSTM-based error correction approach as a fault exclusion methodology that corrects amoralities from multiple EKF filters within federated architecture. The results indicated that the error compensation scheme demonstrates relatively better results when evaluated under different lighting conditions, particularly for straight trajectory flights. However, it struggles to improve performance under foggy weather conditions. This paper has enhanced KF error compensation strategy to design the method 2 hybrid architecture for conducting a comparative study to evaluate the robustness of hybrid architectures under fault scenarios including weather effects. The corrected hybrid VIO estimated states are obtained by following equation-

$$\hat{x}_k = x_k - \epsilon_k \quad (10)$$

where,  $\hat{x}_k$  -denotes corrected method 2 states and  $\epsilon_k = [\epsilon_{k_N}, \epsilon_{k_E}, \epsilon_{k_D}]$  -denotes predicted unmodeled ESKF measurements increments. The forward propagation of the GRU model is adopted using similar data structures in [27] illustrated in Fig. 4.

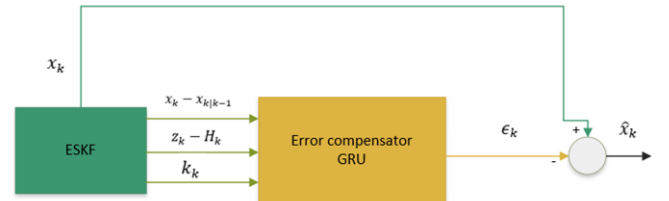


**Fig. 4.** ESKF Error compensation structure for forward propagation

The inputs are composed of ESKF estimating the difference between the state prediction and previous state estimation,  $x_k - x_{k|k-1}$ , the difference between measurement and observation estimation,  $z_k - H_k$  and ESKF estimated gain,  $k_k$ . The predicted ESKF measurement increments are the output of

the GRU model. Regarding GRU architecture, input layers comprise 13 units, with 256 hidden units, a sequence length of 128, a batch size of 64 and a dropout rate of 0.3. The optimizer and lose function remain similar.

#### D. Method 3

The influence of Kalman gain on noise statistics of KF in a dynamic environment inspired the exploration of the hybrid fusion architecture where GRU is employed to predict Kalman gain, aiming to mitigate state nonlinearity. KalmanNet [31] is a popular hybrid architecture that uses the Extended Kalman filtering fusion approach. In situations with complex system dynamics and limited available knowledge, especially limited visual features extracted from the environment, employing an NN-based K-gain prediction is advantageous. This approach enables the updating of KF measurements, overcoming the limitations of model mismatch and nonlinearities imposed by parametric models such as assuming Gaussian noise. Hence, this paper has adopted this phenomenon on the VIO system and compared it with the other two hybrid methods. The GRU-based K-gain prediction follows the ESKF based fusion strategy stated in Section II (A). The prediction of method 3 remains unchanged as the prior estimate,  $x_k$  for current state is computed from the previous posterior state,  $x_{k|k-1}$ . In the update step, the fusion strategy uses the new observation  $y_k$  to compute current state posterior  $x_k$  from previously estimated prior  $x_{k|k-1}$  using innovation  $z_k - H_k$  and predicted Kalman gain,  $k_k$ .

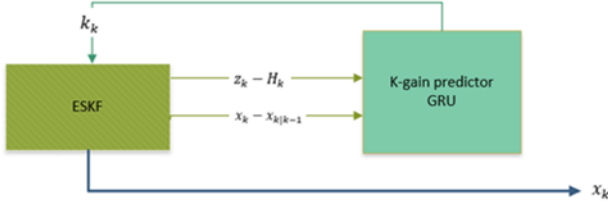


**Fig. 5.** Kalman gain  $k_k$  predictor structure for forward propagation.

The high-level forward propagation structure is illustrated in Fig. 5. Later, the predicted  $k_k$  is replaced in equation 8 which estimates k-gain to update ESKF states and covariance in measurement step. The input layers consist of the difference between measurements and observations,  $z_k - H_k$ ; the difference between posterior state and prior state estimate,  $x_k - x_{k|k-1}$  and the output layer is predicted Kalman gain,  $k_k$ .

The GRU architecture employed in this architecture differs from the GRU architecture utilized in KalmanNet [31] and follows a layout similar to other hybrid VIO systems discussed in the paper, with distinct hyperparameters. The GRU model has a single layer with 128 neurons and 64 sequence length input. The model is trained with a batch size of 128, a dropout rate of 0.4 and a sigmoid activation function. The optimizer and lose function remain similar.

### III. EXPERIMENTAL SETUP AND DATA GENERATION

The section outlines the experiments designed to evaluate the strengths and limitations of the proposed comparative study on

three hybrid approaches. The experimental setup is described first which is used to generate training and testing datasets to conduct the numerical analysis and performance evaluation on each hybrid VIO architectures.

#### A. Experimental Setup

The proposed comparative analysis is implemented and evaluated in MATLAB incorporating the Unreal engine to simulate the data collection process for UAVs. The simulated synthetic datasets encompass a variety of challenging fault conditions including illumination variation, rapid motion, motion variance, flight dynamics, and less texture environments under different weather conditions. The experimental setup for the proposed comparative study is based on the simulation setup used in our previous work with similar configuration parameters for the monocular camera and IMU sensor. Nevertheless, to simulate synthetic weather conditions including partly sunny, fog, cloud, and rain ‘Simulation 3D Scene Configuration’ block in MATLAB 2022a has been updated with the parameters listed in Table 1.

TABLE I  
SPECIFICATION OF THE ‘SIMULATION 3D SCENE CONFIGURATION’ BLOCK IN MATLAB 2022A

| Weather Selection |                     |                     |                    |
|-------------------|---------------------|---------------------|--------------------|
| Sunny             | Sun altitude<br>90  | Sun Azimuth<br>180  |                    |
| Cloud             | Sun altitude<br>90  | Sun Azimuth<br>180  |                    |
|                   | Cloud opacity<br>40 | Cloud speed<br>80   |                    |
| Fog               | Fog Density<br>80   | Sun altitude<br>-45 |                    |
|                   |                     | Sun Azimuth<br>270  |                    |
| Rain              | Rain Density<br>30  | Sun altitude<br>90  | Sun Azimuth<br>180 |
|                   |                     | Cloud opacity<br>40 | Cloud speed<br>80  |

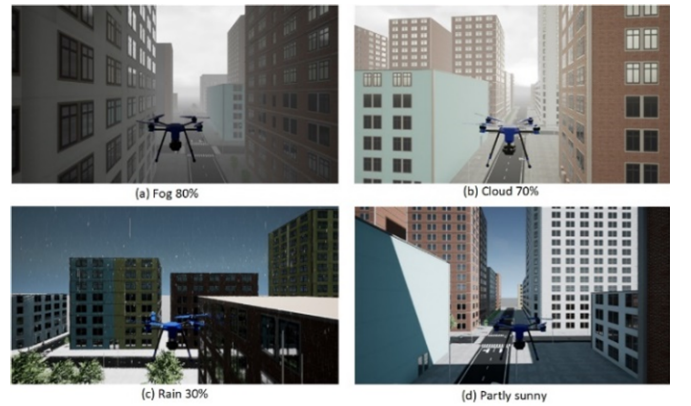


**Fig. 6** Examples of simulated environmental conditions are as follows- (a) limited field of view during 80% fog, (b) texture-less wall during 70% cloud, (c) illumination changes during 30% rain and (d) shadows and texture-less wall during partly sunny used in testing and training datasets.

In urban operation scenarios surrounded by buildings, VIO utilizes a camera to capture visual features from environmental structures (tall buildings, lampposts, trees), providing position information.

TABLE II  
CHARACTERISTICS OF TRAINING TRAJECTORIES

| Training Sequence | Turn followed by the featureless wall. | Skyview with turn | White Rooftop view | Shadow of building on a building | Illumination change | Flying nearby objects (buildings) | Shadow of buildings and trees | Take off with altitude variation. |
|-------------------|----------------------------------------|-------------------|--------------------|----------------------------------|---------------------|-----------------------------------|-------------------------------|-----------------------------------|
| 1                 | ✓                                      | ✓                 |                    | ✓                                | ✓                   |                                   | ✓                             | ✓                                 |
| 2                 |                                        |                   | ✓                  | ✓                                | ✓                   | ✓                                 | ✓                             | ✓                                 |
| 3                 |                                        | ✓                 |                    | ✓                                | ✓                   | ✓                                 | ✓                             | ✓                                 |
| 4                 |                                        |                   |                    | ✓                                | ✓                   | ✓                                 | ✓                             |                                   |
| 5                 |                                        |                   |                    | ✓                                | ✓                   |                                   | ✓                             |                                   |
| 6                 | ✓                                      |                   | ✓                  | ✓                                | ✓                   | ✓                                 | ✓                             | ✓                                 |
| 7                 |                                        | ✓                 |                    | ✓                                |                     | ✓                                 | ✓                             |                                   |
| 8                 |                                        |                   |                    | ✓                                | ✓                   |                                   | ✓                             |                                   |
| 9                 | ✓                                      | ✓                 | ✓                  | ✓                                |                     |                                   | ✓                             | ✓                                 |
| 10                |                                        |                   | ✓                  | ✓                                |                     | ✓                                 | ✓                             |                                   |
| 11                | ✓                                      |                   |                    | ✓                                | ✓                   | ✓                                 | ✓                             | ✓                                 |
| 12                |                                        | ✓                 | ✓                  | ✓                                |                     | ✓                                 | ✓                             |                                   |

The MATLAB simulator interfaces with QgroundControl software to generate multiple trajectories that combine different failure modes and fault conditions, providing control over the selection of single and combined faults. This capability helps to conduct the analysis of the strengths and limitations of different hybrid approaches implemented on VIO systems under complex weather effects and lighting conditions, which is the focus of the paper. Fig. 6 illustrates different fault conditions in different weather effects used in this paper.

#### B. Training Datasets

Regarding the training of the GRU model, 12 sets of trajectories are utilized encompassing environmental fault conditions, flight dynamics, and lighting variation e.g. sunset, evening, midday, and sunrise. Each trajectory spans 15 minutes and features various weather conditions such as fog, sunny, rain, and clouds. A summary of the training trajectories containing fault conditions is given in Table 2. Since the paper utilizes three different hybrid methods, the GRU model employed on each hybrid VIO is trained separately using the data format explained in Section 2.

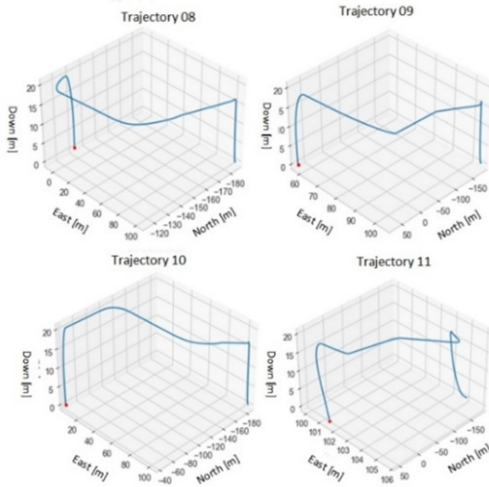


Fig. 7. Examples of trajectories used for training purposes.

For simplicity, the pre-trained GRU model used in previous work is adopted for method 1. Additionally, the GRU model

utilized in each hybrid VIO is trained separately with the 12 distinct datasets, each tailored to the mentioned weather and lighting conditions, following the structures explained in Section 2. The extraction of visual features using an ORB-feature detector by VO is firstly adapted to produce relative pose measurements for the fusion strategy. The IMU measurements utilized in this work have been adopted from previous work. These experiments are conducted in Python, leveraging relevant libraries compatible with TensorFlow for implementation. Fig. 7 illustrates examples of training trajectories used to train the GRU model for each hybrid algorithm.

#### C. Evaluation Metrics

In order to assess the performance of the proposed comparative analysis, the Root Mean Square errors (RMSE), horizontal errors, and 95th percentile are calculated and compared with the three hybrid VIO algorithms. To analyze the behavior of each algorithm and error distribution over failure modes, the whole trajectories visualization plots have been utilized with ground truth. The RMSE calculation for each axis is as follows-

$$RMSE = \sqrt{\sum_{i=1}^N \frac{(\hat{P}_i - P_i)^2}{N}} \quad (11)$$

Where,  $\hat{P}_i$  and  $P_i$  are the predicted value and ground truth along with the N, E, and D position.

RMSE for 3D error is as follows,

$$RMSE_{3D} = \sqrt{\sum_{i=1}^N \frac{((P_{N_i} - P_{N_i})^2 + (P_{E_i} - P_{E_i})^2 + (P_{D_i} - P_{D_i})^2)}{N}} \quad (12)$$

RMSE for horizontal error is as follows-

$$RMSE_{Horizontal} = \sqrt{\sum_{i=1}^N \frac{((P_{N_i} - P_{N_i})^2 + (P_{E_i} - P_{E_i})^2)}{N}} \quad (13)$$

where,  $P_{N_i}$ ,  $P_{E_i}$  and  $P_{D_i}$  are predicted NED position and  $P_{N_i}$ ,  $P_{E_i}$  and  $P_{D_i}$  are the ground truth positioning values from MATLAB incorporated Unreal Engine simulator in NED coordinates.

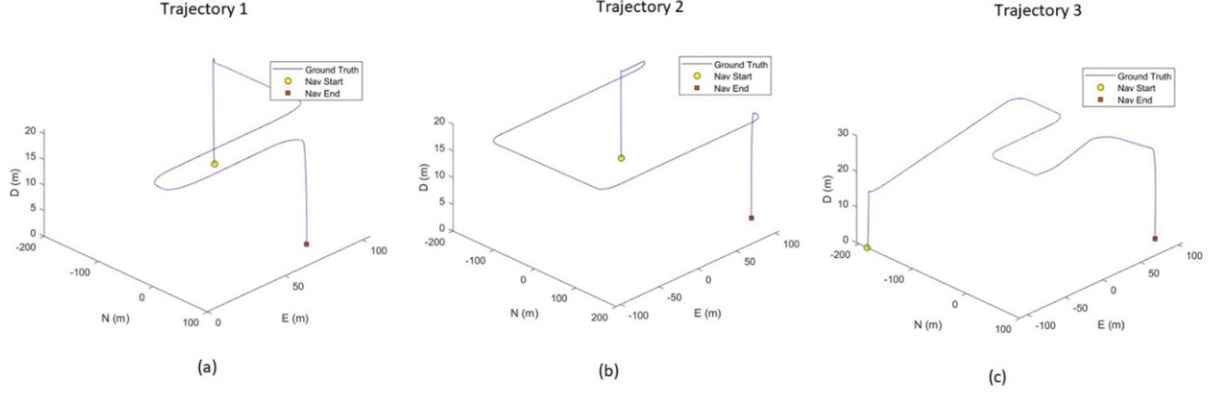


**Fig. 8.** 3D view of test trajectories used for experiments (a) Trajectory 1 for single fault condition, (b) Trajectory 2 for combined fault conditions with weather effects, (c) Trajectory 3 for complex fault unseen scenarios

#### IV. PERFORMANCE EVALUATION

To assess the strengths and limitations of the hybrid VIO algorithms encompassing the purpose of the research, three types of experiments are conducted using the three testing datasets, shown in Fig 8. These trajectories are generated using MATLAB incorporated Unreal engine simulator with a US city scenario which recreates urban environments with various fault conditions including illumination variation, motion variation, altitude variation, no-texture, and weather effects. The details of the three experiments covering fault scenarios with three specific trajectories are described in Table 3.

TABLE III

THE DEFINED EXPERIMENTS COVERING SCENARIOS.

| # Trajectory | Scenarios                                      | Fault type                                                                                                                                                                         |
|--------------|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1            | Single fault condition                         | 1. Shadows of objects<br>2. White Rooftop<br>3. UAV Turn                                                                                                                           |
| 2            | Combined fault conditions with weather effects | 1. Turn with features-wall.<br>2. Turn with sky view.<br>3. Shadows of objects<br>4. Take off with different altitudes                                                             |
| 3            | Complex fault with unseen scenarios            | 1. Turns with featureless walls.<br>2. Turns with sky-view.<br>3. Shadows of objects<br>4. UAV diagonal rotation<br>5. Altitude variation<br>6. Take-off with different altitudes. |

The performance analysis has been categorized based on navigation scenario types that provides an insight into the resilience capabilities of each hybrid VIO schemes in mitigation of visual failure modes. Finally, performance comparisons with state-of-the-art learning-based VO systems are conducted on EuRoc dataset to evaluate resilience under complex fault conditions e.g. darkness, rapid motion, and low-texture environments.

##### A. Single Fault Condition

The purpose of this experiment is to evaluate the effectiveness and limitations of three hybrid VIO systems in managing single-fault conditions and mitigation of failure modes associated with at a time with trajectory 1. During the experiment distance of 702 m is covered over a duration of 152 s. Notably, this experiment is conducted under sunny weather conditions.

Fig. 9 shows the behaviour of 3D trajectories obtained from

three hybrid VIO architectures with the integration of GRU and ESKF refers as method 1, method 2 and method 3. These result trajectories are observed under single fault conditions where the presence of a white rooftop view, as illustrated in Fig. 10 leads to deviation in the performance after the third turn. Notably, white rooftop triggering feature extraction errors and feature tracking errors failure modes within the time interval of 113-140 s. The method 1 effectively mitigates the failure modes and follows closely to the ground trajectory. However, method 2 shows performance degradation while mitigating failure modes like feature tracking errors and feature extraction errors in this experiment under sunny weather conditions.

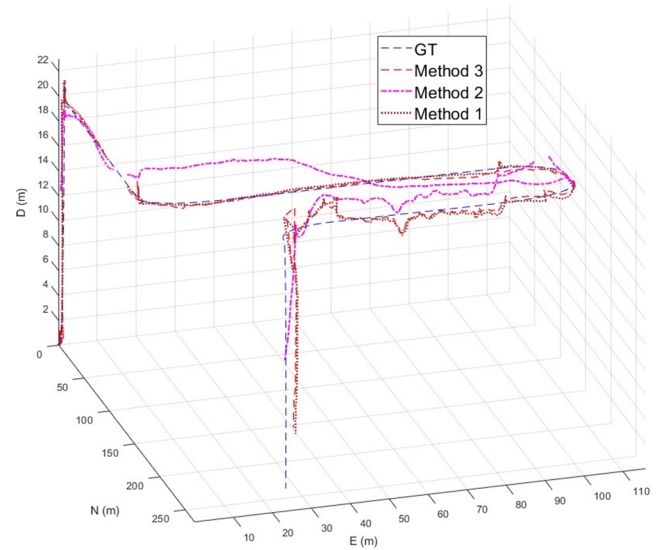


**Fig. 9.** 3D trajectory comparison of three hybrid algorithms denoted as method 1, method 2 and method 3 under single fault condition.

Fig. 11 illustrates the error distribution of each hybrid algorithm relative to ground truth trajectory 1. It is notable that the error along the E coordinate significantly influences overall performance, primarily due to the presence of the featureless rooftop view. However, this effect appears to be mitigated effectively by the method 1 algorithm compared to other hybrid

approaches. The efficacy of the Method 1 is particularly noteworthy in this context as it addresses faults before fusion, thereby preventing the ESKF model from being significantly impacted by the substantial nonlinearity introduced by sensor anomalies. Conversely, the performance of the K-gain prediction mechanism in the E position appears to be less favourable, presumably due to its inability to mitigate the imperfections in the model caused by corrupted sensor measurements.

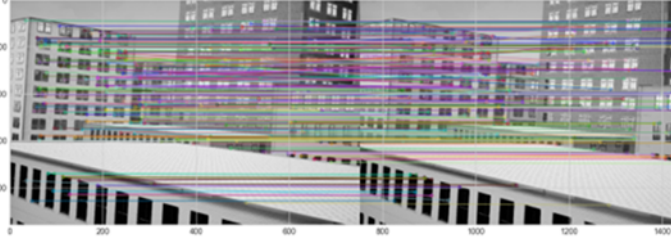


**Fig. 10.** Example of feature tracking errors and feature mismatch failure modes due to the presence of white rooftop view at 115s

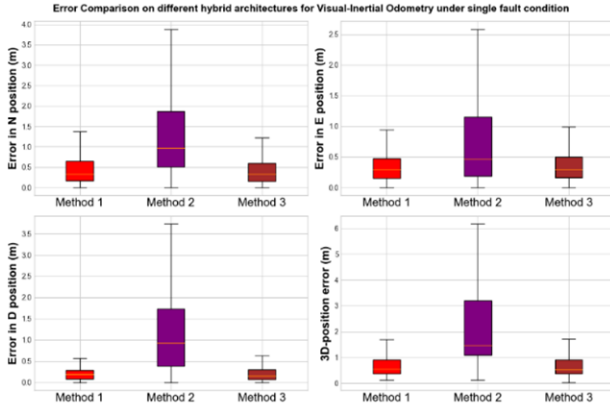


**Fig. 11.** Position error statistics for adopted three hybrid architectures using trajectory 1.

Table 4 summarizes the qualitative results of the hybrid architectures with respect to two reference systems for easy comparison. It is noticed that the influence of minimizing drift with the VO measurement error compensator mechanism is prominent and showed 77.4% and 75.5% improvements in horizontal positioning (95th percentile) compared to the VO and ESKF-VIO reference systems respectively. The prominent failure modes occurring in the horizontal axis when UAV takes turns include feature tracking errors, feature mismatch errors and data association errors exacerbated by factors such as abrupt changes in the environment, occlusion of visual features and rapid motion. The hybrid algorithms are effectively

mitigating these failure modes and improve overall horizontal positioning performance by 85.5%, 70.2%, and 84.1% for method 1, method 2 and method 3 compared to traditional ESKF-VIO respectively.

Fig. 12 illustrates the RMSE in position estimation for each algorithm with two reference systems of VO and ESKF-VIO. The observed performance variance indicates at 60 s, coinciding with the UAV's initial turn, attributable to the limited field of view. This effect persists until the second turn at 85 s, where objects (buildings, lampposts, and trees) are situated far away. The presence of such fault conditions triggers feature extraction errors and feature location errors failure modes which are effectively mitigated by hybrid algorithms. Nevertheless, method 2 has shown degradation performance compared to other two methods in such scenarios. Furthermore, The RMSE position errors from 135-153s are higher where 95th percentile positions are reduced to 81.7%, 20%, 47.2% for VO-error compensation mechanisms, 10%, 17%, 35.6% for KF-error compensation, 17.9%, 34.2%, 59.4% for K-gain predictor concerning ESKF-based VIO fusion along with N, E, and D position. It is observed that the method 1 has shown better performance under single-fault condition while mitigating failure modes including feature extraction errors, feature tracking errors, feature mismatch, data association errors and feature location errors.

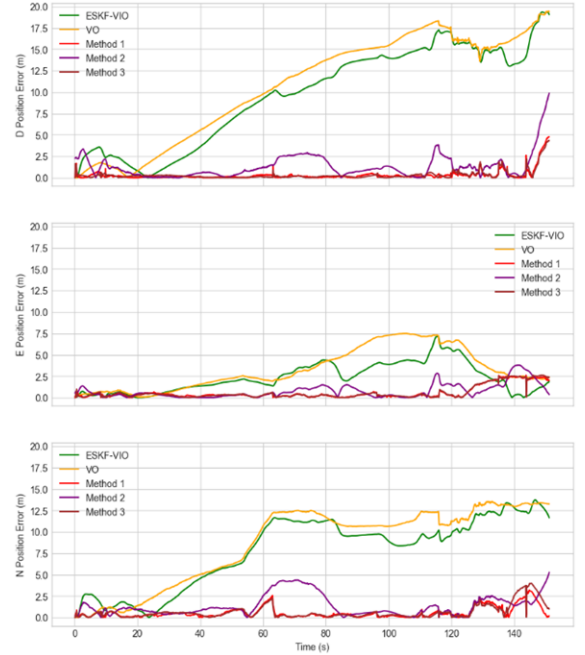


**Fig. 12.** Position error in the presence of a single fault condition for Trajectory 1

TABLE IV  
POSITIONING PERFORMANCE FOR TRAJECTORY 1 UNDER SINGLE FAULT CONDITION.

| Techniques | RMSE (m) |      |       |       | 95 <sup>th</sup> Percentile (m) |      |       |       | Horizontal RMSE (m) | Horizontal 95 <sup>th</sup> Percentile (m) |
|------------|----------|------|-------|-------|---------------------------------|------|-------|-------|---------------------|--------------------------------------------|
|            | N        | E    | D     | 3D    | N                               | E    | D     | 3D    |                     |                                            |
| VO         | 9.7      | 4.03 | 13.36 | 16.24 | 19.5                            | 7.49 | 13.55 | 23.65 | 10.53               | 13.94                                      |
| ESKF-VIO   | 8.02     | 2.26 | 11.72 | 14.24 | 18.14                           | 4.85 | 11.83 | 21.43 | 9.27                | 12.82                                      |
| Method 1   | 0.95     | 0.95 | 0.65  | 1.43  | 1.35                            | 2.31 | 2.71  | 4.22  | 1.32                | 3.14                                       |
| Method 2   | 0.99     | 0.77 | 1.12  | 1.66  | 3.2                             | 2.8  | 3.82  | 4.85  | 2.7                 | 4.27                                       |
| Method 3   | 1.08     | 2.28 | 0.72  | 2.03  | 1.46                            | 2.20 | 3.82  | 5.16  | 1.30                | 3.24                                       |

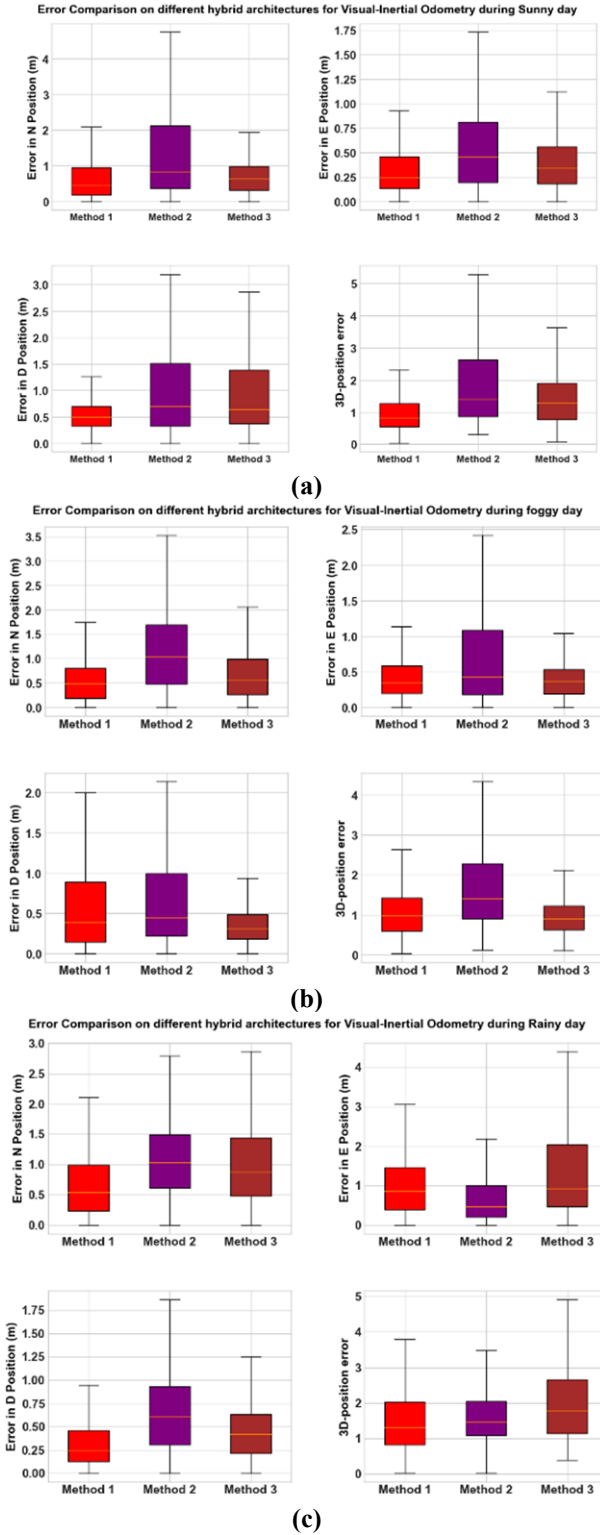


**Fig. 13.** Position error statistics for adopted three hybrid architectures using trajectory 2 under weather effects as follows (a) sunny weather, (b) foggy weather and (c) rainy weather.

#### B. Combined Fault Conditions with Weather Effects

The purpose of this experiment is to evaluate the effectiveness and limitations of three hybrid VIO systems followed by method 1, method 2 and method 3 in managing combined fault conditions and associated failure modes alongside weather effects using

trajectory 2. The experimental environment includes a combination of tall and short buildings with shadows of the environmental objects (e.g. buildings, trees), turn with feature-less wall, turn with skyview, shadow of tall buildings on building and take-off with different altitudes. Three experiments are conducted with a similar trajectory under three weather conditions such as partly sunny, rain and fog. During the experiment distance of 966 m was covered within a duration of 303 s.

In Fig. 13, the boxplots summarizing the position errors of the three hybrid approaches are presented for each coordinate in NED positions and 3D-overall position, illustrating how the results vary for different approaches. Under partly sunny weather conditions, it is observed that the presence of increased noise measurements during extended flight and shadows (illumination variation) with combined fault conditions limit the method 2 and method 3, providing the erroneous position compared to the method 1. These erroneous measurements resulting from multiple failure modes occurring simultaneously, such as feature location errors, feature tracking errors and feature mismatch errors are not only influenced by the illumination changes but also other fault factors. For instance, when the UAV turns near the featureless wall and Skyview, it triggers feature tracking errors and feature extraction errors failure modes. These additional failure modes contribute to the performance reduction observed in ESKF-VIO as illustrated in Fig. 14. However, the hybrid approaches effectively mitigate the effects introduced by multiple failure modes: method 1 by 96%, method 2 by 91.1%, and method 3 by 92.3%. Apparently, complex scenarios like foggy weather conditions contribute to likely feature extraction errors and feature location errors failure modes due to limited visibility to environmental features, resulting in model uncertainty which is improved by learning the K-gain predictor and achieving better overall 3D position performance in 31.5% enhancement by method 3. Furthermore, rain results in water droplets accumulating on the camera lens, blurring visual features, lighting variation and reflections leading to obstruction of feature tracking and matching. It is noticed that the presence of increased noise measurements due to these factors limits method 2 and method 3 techniques, proving improvement in the overall position for method 1. However, the method 1 has shown better performance in the vertical axis during foggy and rainy weather conditions, removing the drift while landing compared to other approaches.

Fig. 14 illustrates position error comparison based on the adopted three hybrid architectures and benchmark with two reference systems of VO and ESKF-based VIO under sunny, foggy, and rainy weather conditions. During the landing phase, notably in foggy and rainy weather conditions, the RMSE errors in the E and D positions are observed higher. It is worth noting that the performance degradation is noticed after 250 s due to combined fault conditions including UAV turn with sky view and shadow of buildings on a building. These conditions lead to illumination variation and less-texture environmental fault conditions, resulting in feature extraction errors, feature location errors and feature tracking errors failure modes. However, fault conditions induced by weather effects such as reflection, overexposure, less visibility, and lighting variation further complicate the navigation environment. For instance, The KF error compensator mechanism exhibits an improvement of 3% and 2% at horizontal (95th percentile) to VO error compensation

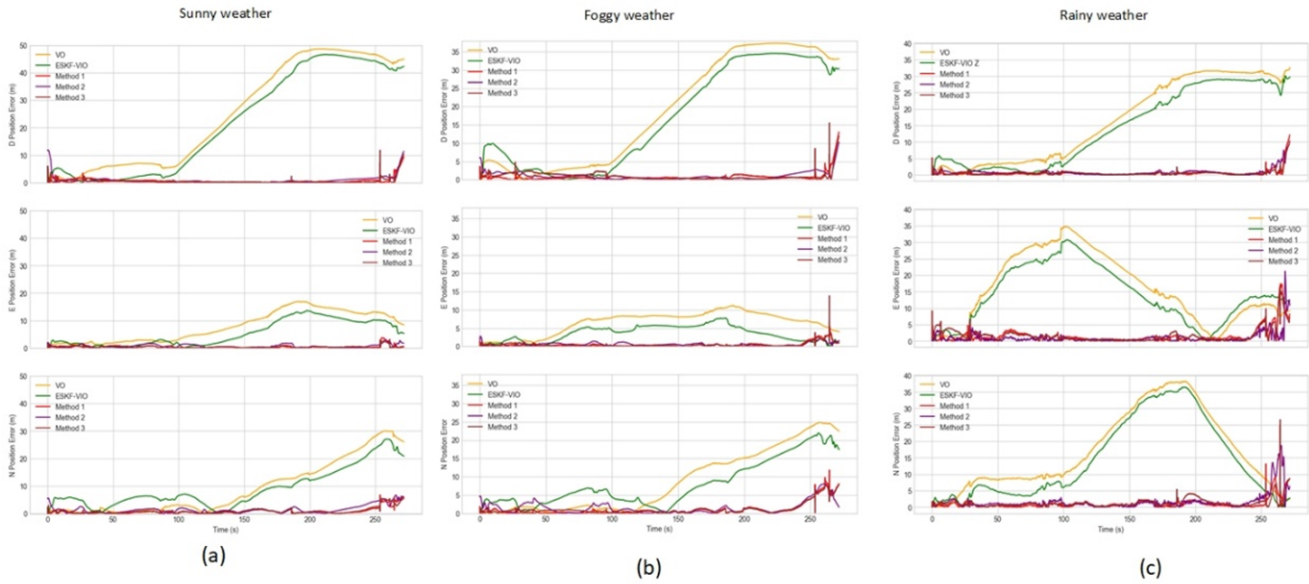


**Fig. 14.** Position error along with each axis in the presence of a single fault condition for trajectory 2 under weather effects as follows (a) sunny weather, (b) foggy weather and (c) rainy weather

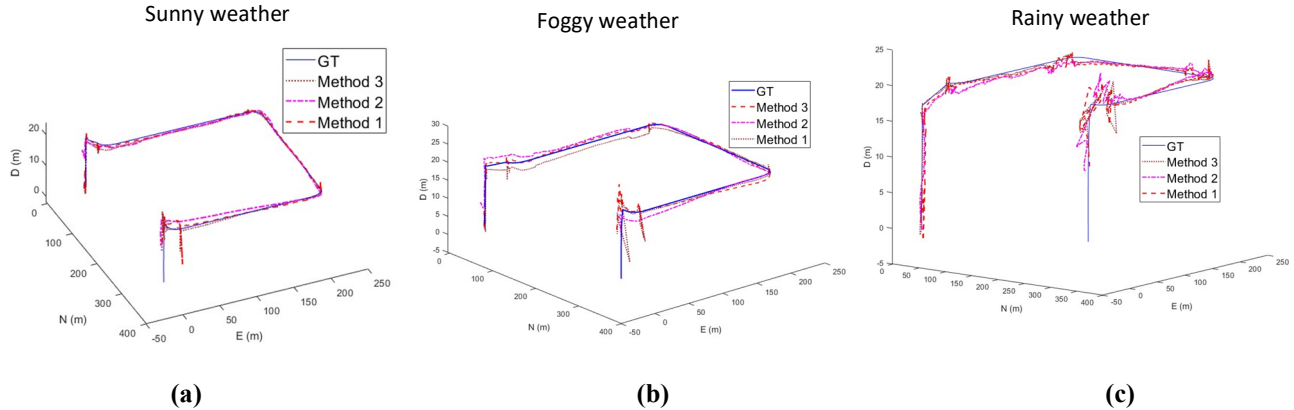


**Fig. 15.** Trajectory comparison of hybrid algorithms for trajectory 2 under weather effects as follows- (a) sunny weather, (b) foggy weather and (c) rainy weather.

and K-gain prediction mechanism under foggy weather conditions. Additionally, the drift observed while landing after 280 s is also reduced up to 2.25 m at the D position (95th percentile).

Table 5, Table 6, and Table 7 summarise the performance of the three hybrid architectures compared to the reference systems concerning RMSE, 95th percentiles, horizontal RMSE and horizontal 95th percentile error under weather effects. It is observed that there is a variation in performance at each algorithm under different fault conditions. During sunny weather, overall horizontal performance (95th percentile) can be improved by 25% and 6% for method 1 and method 3, respectively, compared to method 2. However, method 3 has shown poor performance due to dependency on assumptions and predictive models, sensitivity noise and limitations to adapt complex fault conditions by the predefined model. For instance, in foggy weather conditions characterized by limited visibility, overexposure, and contrast, the overall performance exhibits degradation to 13.40 m in the horizontal position (95th percentile). Furthermore, the method 1 and method 2 schemes overcome such complexity in environments by directly mitigating sensor errors during rainy

weather and measurement errors during sunny and foggy weather. It is observed that compensation of sensor measurements or model measurements is beneficial under this fault condition. Under different weather conditions, it is noted that the prominent failure modes are feature tracking errors and feature extraction errors due to lighting variation, reflection on the surface, speed of raindrops and fewer features extracted from the sky. Such effects introduce erroneous measurements and resulting performance degradation in all hybrid architectures during the landing phase, as discussed. However, the VO error compensation technique shows relatively improved performance, resulting in 27.8%, 30.29%, and 28.95% improvement at the D position (95th percentile) for method 1, method 2, and method 3, respectively, during foggy weather. As per principle, K-gain prediction solely relies on the predefined predictive model, which results in the imperfection of the model. Method 3 can mitigate the fault conditions mentioned above, yet method 1 performs better, boasting a horizontal position error (RMSE) of 2.35 m. Compared to the performance in sunny weather conditions, the drift is a higher position error of 6.59 m, 6.00 m, and 6.99 m (95th percentile) for method 1, method 2, and method 3, respectively.

Fig. 15 illustrates the behaviour of 3D trajectories obtained from three hybrid architectures. These trajectories are evaluated under combined fault conditions across three weather conditions. Notably, deviations in performance are observed at each turn due to the presence of less features. Moreover, during the landing phase, performance degradation is encountered due to complex fault conditions computed by variations in weather. For instance, rainy weather introduces variation in lighting along with raindrops, resulting in failures in feature tracking and extraction simultaneously. An important conclusion about these results can be drawn by comparing the 3D trajectories. The method 1 algorithm shows better performance in mitigating these failure modes along with following the ground truth trajectory 2 under these diverse weather conditions. Furthermore, method 2 indicates better performance in reducing drift and effects of combined associated failure modes, e.g. feature tracking errors, feature location errors and data association errors during landing.

In conclusion, the above analysis underscores the importance of the robustness of hybrid VIO systems with the mitigation of complex and combined visual failure modes, particularly under challenging conditions. The ability of method 1 and method 2 algorithms to adapt diverse fault scenarios and environmental conditions highlights their potential for enhancing navigation accuracy and reliability.

### C. Complex Fault Conditions with Unseen Scenarios

The objective of this experiment is to evaluate the generalization ability of three hybrid VIO systems in managing combined fault conditions, both seen and unseen scenarios within trajectory 3. The experimental setup comprises a diverse scenario, including the combination of tall and short buildings with shadows of the

environmental objects (e.g. buildings, trees), turns with featureless walls, turns with sky-view, shadow of tall buildings on building, diagonal rotation, altitude variation, and take off with different altitudes. The experiment is conducted with trajectory 3 under sunny weather conditions. The flight path deviates from training datasets as it takes a different route while encountering similar patterns throughout the flight. During the experiment, a distance of 738 m was covered within a distance of 230 s.

Fig. 16 presents the behaviour of 3D trajectories obtained from three hybrid approaches. It is observed that the hybrid algorithms demonstrate the ability to generalize effectively across seen and unseen scenarios, closely following the ground truth trajectory. During the takeoff, all hybrid approaches align well with the ground trajectory, successfully adapting to the unseen scenario. However, deviation in performance is observed during the landing phase, particularly due to the diagonally rotated landing of the UAV to an unseen scenario. Notably, among all the hybrid approaches, the method 1 exhibit better performance in mitigating the fault conditions compared to others.

In Fig. 17, the boxplots summarizing the position error of the three hybrid approaches including method 1, method 2 and method 3 across three coordinated of NED positions and 3D-overall position are presented, illustrating how the results vary for different approaches with unseen scenarios. It is observed that the presence of increased noise measurements during extended flight and sunny weather conditions with combined fault conditions limit the method 2 and method 3 mechanism approach, providing the erroneous position compared to method 1. Complex unseen scenarios like diagonally rotation fault conditions contribute to likely feature extraction errors and

TABLE V  
POSITIONING PERFORMANCE FOR TRAJECTORY 2 UNDER SUNNY WEATHER CONDITIONS.

| Techniques | RMSE (m) |      |       |       | 95 <sup>th</sup> Percentile (m) |       |       |       | Horizontal RMSE (m) | Horizontal 95 <sup>th</sup> percentile (m) |
|------------|----------|------|-------|-------|---------------------------------|-------|-------|-------|---------------------|--------------------------------------------|
|            | N        | E    | D     | 3D    | N                               | E     | D     | 3D    |                     |                                            |
| VO         | 12.10    | 7.20 | 23.69 | 27.56 | 11.35                           | 10.50 | 24.09 | 43.47 | 14.08               | 24.80                                      |
| ESKF-VIO   | 10.08    | 4.11 | 21.60 | 24.19 | 34.52                           | 6.79  | 20.42 | 39.01 | 10.89               | 20.45                                      |
| Method 1   | 1.29     | 0.68 | 1.22  | 1.86  | 2.00                            | 0.86  | 4.09  | 4.39  | 1.47                | 4.36                                       |
| Method 2   | 1.28     | 0.51 | 1.28  | 2.13  | 1.79                            | 1.06  | 2.25  | 3.00  | 2.18                | 5.03                                       |
| Method 3   | 1.44     | 0.56 | 1.18  | 1.95  | 1.27                            | 0.85  | 3.67  | 4.59  | 1.56                | 4.05                                       |

TABLE VI  
POSITIONING PERFORMANCE FOR TRAJECTORY 2 UNDER FOGGY WEATHER CONDITIONS.

| Techniques | RMSE (m) |      |       |       | 95 <sup>th</sup> Percentile (m) |       |       |       | Horizontal RMSE (m) | Horizontal 95 <sup>th</sup> percentile (m) |
|------------|----------|------|-------|-------|---------------------------------|-------|-------|-------|---------------------|--------------------------------------------|
|            | N        | E    | D     | 3D    | N                               | E     | D     | 3D    |                     |                                            |
| VO         | 13.16    | 9.47 | 30.97 | 34.95 | 48.26                           | 16.94 | 28.32 | 56.09 | 16.21               | 30.96                                      |
| ESKF-VIO   | 11.38    | 6.94 | 28.60 | 31.56 | 46.61                           | 12.72 | 23.94 | 50.61 | 13.33               | 25.76                                      |
| Method 1   | 2.29     | 0.62 | 1.58  | 2.35  | 1.87                            | 1.37  | 7.03  | 7.51  | 2.13                | 6.32                                       |
| Method 2   | 2.92     | 0.41 | 1.48  | 3.31  | 4.20                            | 1.00  | 8.24  | 6.39  | 2.33                | 5.36                                       |
| Method 3   | 1.82     | 0.76 | 1.29  | 2.87  | 1.46                            | 1.65  | 5.36  | 9.40  | 2.05                | 6.01                                       |

TABLE VII  
POSITIONING PERFORMANCE FOR TRAJECTORY 2 UNDER RAINY WEATHER CONDITIONS.

| Techniques | RMSE (m) |       |       |       | 95 <sup>th</sup> Percentile (m) |       |       |       | Horizontal RMSE (m) | Horizontal 95 <sup>th</sup> percentile (m) |
|------------|----------|-------|-------|-------|---------------------------------|-------|-------|-------|---------------------|--------------------------------------------|
|            | N        | E     | D     | 3D    | N                               | E     | D     | 3D    |                     |                                            |
| VO         | 20.54    | 19.01 | 20.48 | 34.69 | 36.22                           | 10.50 | 24.09 | 43.47 | 27.99               | 39.57                                      |
| ESKF-VIO   | 18.54    | 16.50 | 18.17 | 30.76 | 34.45                           | 6.79  | 20.42 | 39.01 | 24.82               | 36.253                                     |
| Method 1   | 1.53     | 2.70  | 1.27  | 3.45  | 0.74                            | 0.47  | 2.99  | 6.28  | 3.25                | 6.64                                       |
| Method 2   | 3.11     | 2.15  | 1.27  | 3.93  | 4.49                            | 6.61  | 4.17  | 6.43  | 3.60                | 10.29                                      |
| Method 3   | 1.99     | 2.94  | 3.35  | 3.80  | 1.64                            | 6.05  | 7.67  | 7.75  | 3.29                | 7.03                                       |

feature tracking errors, resulting in model uncertainty, which is improved by learning fault patterns in sensor data and achieving better D position performance to 1.28 m RMSE error. However, the performance improvement was shown by method 1 in the 3D position by 19.86% compared to the ESKF- VIO reference system. The N position acts after takeoff when the UAV enters into an unseen scenario. Results from Table 8 indicate method 2 can minimize the erroneous measurements introduced after flight takeoff in the horizontal axis due to that turn with a feature-less wall and variation of altitude at each turn. It is noticed that the presence of increased noise measurements indicates imperfection in the model, thereby limiting the effectiveness of method 3 compared to error compensation techniques. The findings address noise-induced imperfections in measurements and highlight that the error compensation techniques work better, particularly in scenarios involving variations in altitude and no-texture environments.

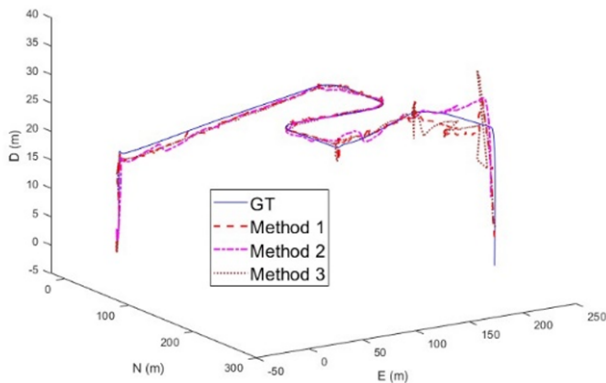


**Fig. 16.** 3D trajectories of hybrid algorithms under complex fault conditions with unseen scenarios for trajectory 3.

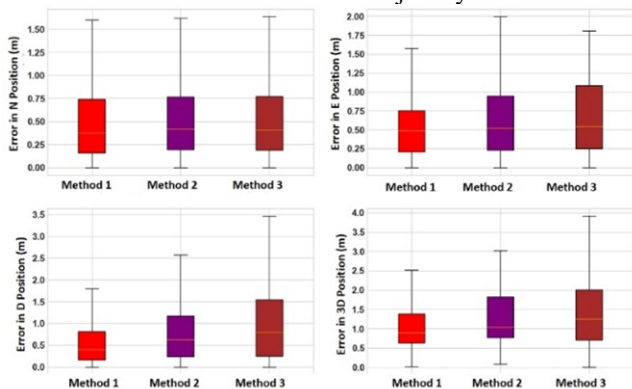


**Fig. 17.** Boxplot summarizing the position error statistics for the three hybrid architectures using trajectory 3.

TABLE VIII

POSITIONING PERFORMANCE FOR TRAJECTORY 3 UNDER UNSEEN SCENARIOS.

| Techniques | RMSE (m) |      |       |       | Horizontal RMSE (m) |
|------------|----------|------|-------|-------|---------------------|
|            | N        | E    | D     | 3D    |                     |
| VO         | 4.04     | 5.13 | 18.11 | 19.25 | 6.55                |
| ESKF-VIO   | 3.77     | 4.85 | 16.00 | 17.15 | 6.12                |
| Method 1   | 1.30     | 3.36 | 1.28  | 3.83  | 3.76                |
| Method 2   | 0.76     | 3.46 | 2.46  | 4.31  | 3.44                |

|          |      |      |      |      |      |
|----------|------|------|------|------|------|
| Method 3 | 1.40 | 3.36 | 1.98 | 4.16 | 3.94 |
|----------|------|------|------|------|------|

#### D. Comparison to Learning-based VIO

This section presents a performance comparison between the adopted hybrid approaches and state-of-the-art learning-based VIO solutions. Given the limited availability of open-source state-of-the-art learning-based VIO solutions, in this study a well-known publicly available EuRoC UAV dataset [34] for evaluation is utilized. To provide insight into the performance of the three hybrid methods explored in this study, the complex sequences, including MH05\_difficult, were selected for benchmarking. MH05\_difficult was collected with Micro Aerial Vehicles (MAV) indoors and contains several fault conditions, such as darkness, illumination variation, motion variation, and motion blur. These factors contribute to various failure modes, including feature extraction errors, feature location errors, feature tracking errors and data association errors. Notably, the dataset was selected as it includes complex scenarios with fault conditions analysed earlier in Section IV using simulated data. To provide the performance reference from the state-of-the-art (SOTA) solutions, three learning-based VIO architectures were selected: End-to-End VIO [12], DeepVIO [38] and Self-VIO [23]. These algorithms were selected based on the availability of experimental results using the selected EuRoC sequence. Since only 3D position performance was provided by the authors of the selected SOTA algorithms, the comparative performance analysis of the proposed methods is limited to this metric.

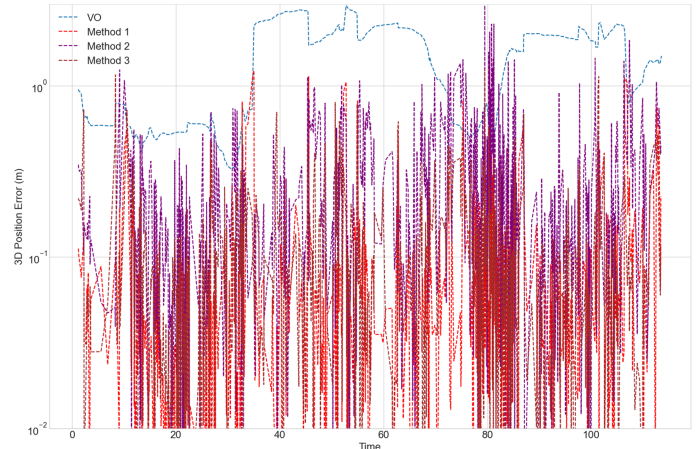


**Fig. 18.** 3D position error for the three hybrid architectures using EuRoC dataset (MH05\_difficult) with complex failure modes.

Fig. 18 represents comparative results for MH05\_difficult dataset. It is observed that three hybrid VIO schemes outperform traditional feature-based VO that struggles due to the environmental complexity present in the data. Difficulties arise from harsh motion dynamics, increased degree of freedom and illumination variation in the indoor environment. However, the images and imu data within the dataset have time discontinuity due to the methodology adopted during the dataset collection process. It has been observed that method 2 struggles to perform compared to others due to intensive nonlinearity present in the dataset. Method 1 outperforms all the hybrid solutions.

Table 9 highlights the comparison results in terms of RMSE with state-of-the-art solutions. Performance degradation in

Method 2 is attributed to the nonlinear dynamics of MAV (rapid motion) and environmental factors (blur and darkness). These additional noise artifacts render the predictions less reliable. SOTA architectures evaluated their solutions by splitting the datasets and reinitializing the starting points as described in [14], [35], [36], [37]. In contrast, this study did not split the dataset; instead, it was processed through the fusion solutions to achieve optimal results. Thus, the hybrid fusion solutions outperform SOTA learning-based architectures.

TABLE IX

POSITIONING PERFORMANCE FOR MH05\_DIFFICULT DATASET FROM EUROC DATASET.

| Techniques          | RMSE (m) |      |      |      | Horizontal RMSE (m) |
|---------------------|----------|------|------|------|---------------------|
|                     | N        | E    | D    | 3D   |                     |
| End-to-End VIO [12] | -        | -    | -    | 1.96 | -                   |
| DeepVIO [38]        | -        | -    | -    | 0.52 | -                   |
| Self-VIO[23]        | -        | -    | -    | 0.29 | -                   |
| Method 1            | 0.34     | 0.29 | 0.28 | 0.66 | 0.49                |
| Method 2            | 0.88     | 0.65 | 0.11 | 1.03 | 0.96                |
| Method 3            | 0.32     | 0.36 | 0.29 | 0.69 | 0.53                |

## V. CONCLUSION

This study proposed and investigated three hybrid architectures implemented on VIO system with integration of GRU and ESKF where GRU is utilized for three different purposes to enhance ESKF performance under complex urban environments. The three hybrid VIO system has been compared based on their performance mitigating visual failure modes and resilience under different weather effects. The evaluation also encompasses generalization ability of each approach to provide insight into their robustness. The presented systematic analysis has shown that method 1 exhibits better performance regarding mitigating fault conditions in sunny, foggy, and rainy weather. Besides, the method 2 effectively mitigates the performance degradation caused by hover turn in less feature environments, thereby reducing feature extraction errors with feature tracking errors failure modes. Therefore, the method 2 works better than method 1 and method 3 approaches in certain scenarios due to their ability to adjust the filter's state estimates by mitigating model errors. This adjustment allows the algorithm to better handle the specific combinational failure mode types occurring during hovering turn, roll turn and landing drift. Conversely, out of all the hybrid approaches, not one method performed consistently well out of the evaluated fault conditions. Challenges remain, particularly in scenarios where multiple fault conditions converge such as diagonal rotation, landing drift or when navigating featureless environments under rainy and foggy weather conditions.

Future Work will focus on implementing the best solution e.g. visual sensor error correction using GRU (method 1) in real-time to evaluate the performance effectiveness and robustness under failure modes like feature extraction errors and feature tracking errors in the urban scenario. Besides, additional diversity in training datasets including complex fault scenarios to enhance the prediction accuracy of the learning based GRU algorithm can be the forward step towards enhancing the adaptability of the system. Incorporating combinational hybrid architectures may complement each other's limitations and

enhance the overall performance under practical operating conditions.

## REFERENCES

- [1] B. Wagstaff, E. Wise, and J. Kelly, "A Self-Supervised, Differentiable Kalman Filter for Uncertainty-Aware Visual-Inertial Odometry," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 2022-July, pp. 1388–1395, 2022, doi: 10.1109/AIM52237.2022.9863270.
- [2] H. Yin, P. X. Liu, and M. Zheng, "Stereo Visual Odometry With Automatic Brightness Adjustment and Feature Tracking Prediction," *IEEE Trans Instrum Meas*, vol. 72, 2023, doi: 10.1109/TIM.2022.3223070.
- [3] C. Fan, J. Hou, and L. Yu, "Large-Scale Dense Mapping System Based on Visual-Inertial Odometry and Densely Connected U-Net," *IEEE Trans Instrum Meas*, vol. 72, 2023, doi: 10.1109/TIM.2023.3250301.
- [4] Y. Fu, S. Wang, Y. Zhai, and X. Zhan, "Visual odometry errors and fault distinction for integrity monitoring," *AerSy*, vol. 3, no. 4, pp. 265–274, Dec. 2020, doi: 10.1007/S42401-020-00062-X.
- [5] D. You, J. Jung, and J. Oh, "Enhancing Low-light Images for Monocular Visual Odometry in Challenging Lighting Conditions," *Int J Control Autom Syst*, vol. 21, no. 11, pp. 3528–3539, Nov. 2023, doi: 10.1007/S12555-023-0378-7.
- [6] X. Bai, W. Wen, and L. T. Hsu, "Degeneration-Aware Outlier Mitigation for Visual Inertial Integrated Navigation System in Urban Canyons," *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2021.3126010.
- [7] L. Yu, E. Yang, B. Yang, Z. Fei, and C. Niu, "A Robust Learned Feature-Based Visual Odometry System for UAV Pose Estimation in Challenging Indoor Environments," *IEEE Trans Instrum Meas*, vol. 72, 2023, doi: 10.1109/TIM.2023.3279458.
- [8] A. Yusefi, A. Durdu, M. F. Aslan, and C. Sungur, "LSTM and filter based comparison analysis for indoor global localization in UAVs," *IEEE Access*, vol. 9, pp. 10054–10069, 2021, doi: 10.1109/ACCESS.2021.3049896.
- [9] L. Xia, D. Meng, J. Zhang, D. Zhang, and Z. Hu, "Visual-inertial simultaneous localization and mapping: Dynamically fused point-line feature extraction and engineered robotic applications," *IEEE Trans Instrum Meas*, 2022, doi: 10.1109/TIM.2022.3198724.
- [10] X. Ban, H. Wang, T. Chen, Y. Wang, and Y. Xiao, "Monocular Visual Odometry Based on Depth and Optical Flow Using Deep Learning," *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2020.3024011.
- [11] W. Sun, Y. Li, W. Ding, and J. Zhao, "A Novel Visual Inertial Odometry Based on Interactive Multiple Model and Multistate Constrained Kalman Filter," *IEEE Trans Instrum Meas*, vol. 73, 2024, doi: 10.1109/TIM.2023.3335514.
- [12] C. Li and S. L. Waslander, "Towards End-to-end Learning of Visual Inertial Odometry with an EKF," *Proceedings - 2020 17th Conference on Computer and Robot Vision, CRV 2020*, pp. 190–197, May 2020, doi: 10.1109/CRV50864.2020.00033.
- [13] Y. Almalioglu, M. Turan, N. Trigoni, and A. Markham, "Deep learning-based robust positioning for all-weather autonomous driving," *Nat Mach Intell*, vol. 4, no. 9, pp. 749–760, Sep. 2022, doi: 10.1038/S42256-022-00520-5.
- [14] Y. X. Tan, M. B. Prasetyo, M. A. Daffa, D. S. Nitin, and M. Meghjani, "Evaluating Visual Odometry Methods for Autonomous Driving in Rain," *IEEE International Conference on Automation Science and Engineering*, vol. 2023-August, Sep. 2023, doi: 10.1109/CASE56687.2023.10260549.

- [15] V. R. F. Miranda, A. M. C. Rezende, T. L. Rocha, H. Azpúrua, L. C. A. Pimenta, and G. M. Freitas, "Autonomous Navigation System for a Delivery Drone," *Journal of Control, Automation and Electrical Systems*, vol. 33, no. 1, pp. 141–155, Feb. 2022, doi: 10.1007/S40313-021-00828-4/TABLES/2.
- [16] I. Ogunrinde and S. Bernadin, "Deep Camera–Radar Fusion with an Attention Framework for Autonomous Vehicle Vision in Foggy Weather Conditions," *Sensors (Basel)*, vol. 23, no. 14, p. 6255, Jul. 2023, doi: 10.3390/S23146255.
- [17] L. Jeelani SAYED and P. Kumar SOURIPALLI, "Autonomous Navigation of Drones Using Explainable Deep Reinforcement Learning in Complex Environments", doi: 10.54808/WMSCI2024.01.44.
- [18] T. Ort, I. Giltschenski, and D. Rus, "Autonomous navigation in inclement weather based on a localizing ground penetrating radar," *IEEE Robot Autom Lett*, vol. 5, no. 2, pp. 3267–3274, Apr. 2020, doi: 10.1109/LRA.2020.2976310.
- [19] V. S. Saravananarajan, R. C. Chen, C. H. Hsieh, and L. S. Chen, "Improving Semantic Segmentation Under Hazy Weather for Autonomous Vehicles Using Explainable Artificial Intelligence and Adaptive Dehazing Approach," *IEEE Access*, vol. 11, pp. 38194–38207, 2023, doi: 10.1109/ACCESS.2023.3251728.
- [20] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, Sep. 2017, doi: 10.1177/0278364917728574/ASSET/IMAGES/LARGE/10.1177\_0278364917728574-FIG20.JPEG.
- [21] F. Ma, J. Shi, Y. Yang, J. Li, and K. Dai, "ACK-MSCKF: Tightly-coupled ackermann multi-state constraint kalman filter for autonomous vehicle localization," *Sensors (Switzerland)*, vol. 19, no. 21, Nov. 2019, doi: 10.3390/s19214816.
- [22] G. Costante and M. Mancini, "Uncertainty estimation for data-driven visual odometry," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1738–1757, Dec. 2020, doi: 10.1109/TRO.2020.3001674.
- [23] Y. Almalioğlu, M. Turan, R. U. Saputra, P. P. B. De Gusmão, A. Markham, and N. Trigoni, "SelfVIO: Self-supervised deep monocular Visual-Inertial Odometry and depth estimation," *Neural Networks*, vol. 150, pp. 119–136, 2022, doi: 10.1016/j.neunet.2022.03.005.
- [24] M. F. Aslan, A. Durdu, and K. Sabanci, "Visual-Inertial Image-Odometry Network (VIIONet): A Gaussian process regression-based deep architecture proposal for UAV pose estimation," *Measurement*, vol. 194, p. 111030, May 2022, doi: 10.1016/J.MEASUREMENT.2022.111030.
- [25] S. A. Negru, P. Geragersian, I. Petrunin, and W. Guo, "Resilient Multi-Sensor UAV Navigation with a Hybrid Federated Fusion Architecture," *Sensors* 2024, Vol. 24, Page 981, vol. 24, no. 3, p. 981, Feb. 2024, doi: 10.3390/S24030981.
- [26] T. E. Tabassum, Z. Xu, I. Petrunin, and Z. A. Rana, "Integrating GRU with a Kalman Filter to Enhance Visual Inertial Odometry Performance in Complex Environments," *Aerospace* 2023, Vol. 10, Page 923, vol. 10, no. 11, p. 923, Oct. 2023, doi: 10.3390/AEROSPACE10110923.
- [27] M. Villalonga Torres, T. Elmi Tabassum, I. Petrunin, and A. Tsourdos, "Hybrid Multi-Sensor Navigation System With Uncertainty Correction for GNSS-Denied Environments," *AIAA SCITECH 2024 Forum*, Jan. 2024, doi: 10.2514/6.2024-2799.
- [28] S. Kim, I. Petrunin, and H. S. Shin, "A Review of Kalman Filter with Artificial Intelligence Techniques," *Integrated Communications, Navigation and Surveillance Conference, ICNS*, vol. 2022-April, 2022, doi: 10.1109/ICNS54818.2022.9771520.
- [29] S. Feng, X. Li, S. Zhang, Z. Jian, H. Duan, and Z. Wang, "A review: state estimation based on hybrid models of Kalman filter and neural network," *Systems Science & Control Engineering*, vol. 11, no. 1, p. 2173682, Dec. 2023, doi: 10.1080/21642583.2023.2173682.
- [30] D. J. Jwo, A. Biswal, and I. A. Mir, "Artificial Neural Networks for Navigation Systems: A Review of Recent Research," *Applied Sciences* 2023, Vol. 13, Page 4475, vol. 13, no. 7, p. 4475, Mar. 2023, doi: 10.3390/AP13074475.
- [31] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. Van Sloun, and Y. C. Eldar, "KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022, doi: 10.1109/TSP.2022.3158588.
- [32] P. Geragersian, I. Petrunin, W. Guo, and R. Grech, "An INS/GNSS fusion architecture in GNSS denied environments using gated recurrent units," *AIAA Science and Technology Forum and Exposition, AIAA SciTech Forum 2022*, 2022, doi: 10.2514/6.2022-1759.
- [33] N. Shaukat, A. Ali, M. J. Iqbal, M. Moinuddin, and P. Otero, "Multi-sensor fusion for underwater vehicle localization by augmentation of rbf neural network and error-state kalman filter," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–26, Feb. 2021, doi: 10.3390/s21041149.
- [34] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, Sep. 2016, doi: 10.1177/0278364915620033/ASSET/IMAGES/LARGE/10.1177\_0278364915620033-IMG2.JPEG.
- [35] T. Qin, P. Li, and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018, doi: 10.1109/TRO.2018.2853729.
- [36] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.
- [37] M. F. Aslan, A. Durdu, A. Yusefi, and A. Yilmaz, "HVIONet: A deep learning based hybrid visual-inertial odometry approach for unmanned aerial system position estimation," *Neural Networks*, vol. 155, pp. 461–474, Nov. 2022, doi: 10.1016/J.NEUNET.2022.09.001.
- [38] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised Deep Learning of Monocular Visual Inertial Odometry using 3D Geometric Constraints," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6906–6913, Nov. 2019, doi: 10.1109/IROS40897.2019.8968467.



**DSpace**

<https://dspace.org/>

## **A comparative analysis of hybrid sensor fusion schemes for visual–inertial navigation**

**Tabassum, Tarafder Elmi; Petrunin, Ivan; Rana, Zeeshan A.**

**2025**

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/tim.2025.3555758>

Tabassum TE, Petrunin I, Rana ZA. (2025) A comparative analysis of hybrid sensor fusion schemes for visual–inertial navigation. IEEE Transactions on Instrumentation and Measurement, Volume 74, 2025, Article number 8506415