

Article

Sequence-to-Sequence Remaining Useful Life Prediction of the Highly Maneuverable Unmanned Aerial Vehicle: A Multilevel Fusion Transformer Network Solution

Shaojie Ai ^{1,2} , Jia Song ^{1,2,*}  and Guobiao Cai ^{1,3}

¹ School of Astronautics, Beihang University, Beijing 100191, China; aishaojie@buaa.edu.cn (S.A.); cgb@buaa.edu.cn (G.C.)

² Aerospace Crafts Technology Institute, Beihang University, Beijing 100191, China

³ Key Laboratory of Spacecraft Design Optimization and Dynamic Simulation Technologies of Ministry of Education, Beihang University, Beijing 100191, China

* Correspondence: songjia@buaa.edu.cn

Abstract: The remaining useful life (RUL) of the unmanned aerial vehicle (UAV) is primarily determined by the discharge state of the lithium-polymer battery and the expected flight maneuver. It needs to be accurately predicted to measure the UAV's capacity to perform future missions. However, the existing works usually provide a one-step prediction based on a single feature, which cannot meet the reliability requirements. This paper provides a multilevel fusion transformer-network-based sequence-to-sequence model to predict the RUL of the highly maneuverable UAV. The end-to-end method is improved by introducing the external factor attention and multi-scale feature mining mechanism. Simulation experiments are conducted based on a high-fidelity quad-rotor UAV electric propulsion model. The proposed method can rapidly predict more precisely than the state-of-the-art. It can predict the future RUL sequence by four-times the observation length (32 s) with a precision of 83% within 60 ms.



Citation: Ai, S.; Song, J.; Cai, G.

Sequence-to-Sequence Remaining Useful Life Prediction of the Highly Maneuverable Unmanned Aerial Vehicle: A Multilevel Fusion Transformer Network Solution.

Mathematics **2022**, *10*, 1733. <https://doi.org/10.3390/math10101733>

Academic Editors: Daniel-Ioan Curiac and Alfonso Niño

Received: 31 March 2022

Accepted: 17 May 2022

Published: 18 May 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: remaining useful life; sequence-to-sequence prognostics; transformer network; unmanned aerial vehicle; lithium-polymer battery

MSC: 68T40

1. Introduction

The quad-rotor UAV (QUAV) has vertical take-off and landing capability, air hovering capability, payload-carrying capability, and autonomous or remote control capability [1]. The distinctive advantages above make it widely used in agriculture, transportation, and other civilian domains. At the same time, it can casually switch several flight modes at low altitude, in a narrow, dark, or rough environment. Thus, it has the potential to replace human beings to accomplish dangerous tasks. As the environment changes, the drone system is subject to complex disturbances, physical limitations, and flight constraints [2]. The highly maneuverable QUAV must overcome these difficulties to ensure the ability to respond rapidly in various working conditions [3]. To best utilize QUAV's mission capability, the remaining useful life (RUL) has become a necessary measurement standard for mission planning and assessment of residual flight capability. Additionally, to account for endurance and low mass, lithium-polymer (Li-PO) batteries with a high energy density are generally used to power the drone. Hence, the RUL prediction based on the Li-PO battery has attracted extensive worldwide attention and becomes a research hotspot in the field of UAV fault prognosis and health management (PHM) [4].

The RUL of the Li-PO battery cannot be directly observed and measured. It must be calculated from correlated measured elements [5], resulting in considerable uncertainty in its estimation and prediction. Moreover, the flight plan is easy to make overly conservative, which is not conducive to the use of the highly maneuverable QUAV's mission capability. The existing RUL prediction technology is mainly divided into the model-based method

and data-driven method. The model-based method serves as the abstraction for making probabilistic statements about questions of interest, i.e., Bayesian-based methods [6–8]. While the prior information can be fully used, it relies heavily on the prediction model based on the degradation mechanism of the system. Researchers have made improvements in modeling the probability distribution of the battery discharge process through the solvable mathematical model [9]. However, little attention has been paid to the influence of the strong coupling and high nonlinearity, which are significant characteristics of the UAV, on modeling. In particular, the model mismatch in the frequent maneuvers will further increase the uncertainty and lead to application difficulties. Contrary to the model-based method, the data-driven method makes it possible to construct a model-free prediction network by mining the features from the data flow, thus with obvious advantages [10]. Both classical machine learning and deep learning methods are included in the data-driven method [11]. The classical machine learning method trains classifiers with the best performance through advanced data-processing technology and powerful algorithm technology [12]. In order to guarantee the precision of the algorithm, feature extraction and pattern recognition are performed successively and separately. The kernel principal component analysis and the hybrid neural network were combined by Miao et al. for predicting the aircraft engine RUL [13]. Sarkar et al. [14] carried out multicollinearity analysis screening before the sensor feature enters the fully connected neural network. However, the sensitivity of the extracted feature is subject to the expert experience and expertise. The extraction process is often cumbersome and time-consuming due to the high-precision demands. Therefore it is not conducive to real-time tasks, i.e., online prediction. On the contrary, the deep learning method independently explores the data pattern through the neural networks to complete the learning of classification and prediction tasks, with an end-to-end data fitting capacity.

In the field of prediction, the top deep learning baseline methods are primarily: the long short-term memory network (LSTM-based) [15,16], the temporal convolutional network (TCN-based), and the transformer network (TF-based). The LSTM-based method takes the recurrent neural network as the basic framework and introduces the hidden state storage mechanism to learn the sequential representation of historical data. Zhao et al. [17] designed the bidirectional gated recurrent unit network to weight the different local features to predict the current state of the machine. Liang et al. [18] proposed a multilevel network based on the LSTM to predict the future readings of geo-sensors, whereas the sequential processing of the observations leads to the inadequate representation of the early temporal features. The TCN-based method achieves the parallel calculation using the convolutional network architecture and can use all historical information [19]. Song et al. [20] established a TCN-based structure with a feature-weighted optimization, achieving the weight of multi-sensor data at different times to a certain extent during the RUL prediction process. However, these complex weighting operations lead to omissions, such as the correlation among sensors. Furthermore, neither of the above two methods can perform the sequence-to-sequence prediction, as they require the input and output to have the same duration. The transformer network (TF), as a breakthrough in deep learning, has been embraced in a variety of areas such as natural language processing [21], computer vision [22], trajectory forecasting [23], etc. Research has shown that TF is more effective in the aforementioned areas where traditional deep learning methods are usually used. TF processes time sequential data in parallel through positional encoding and the self-attention mechanism, which develops the feature-extraction ability and may process some missing observation data. It should be noted that the joint structure of the encoder and decoder endows it with the ability of sequence-to-sequence prediction. The prediction model proposed by Mo et al. lacks a decoder, so it only has a nonlinear regression function [24].

The deep learning method has also made many achievements in the QUAV RUL prediction. The application of the Bayesian neural network was explored in [25], but its complete dependency on the voltage makes the precision low. The driving capacity of the battery is power load dependent, and the RUL is related to the discharge pattern, payload, and flight mode. More features may better reflect RUL changes, but may also bring redundancy and burden. One of the problems with the deep learning model design is how to better integrate multi-source features [14]. When the QUAV changes maneuver quickly

and significantly, the RUL also changes frequently, emphasizing higher requirements for the speed and precision of the prediction algorithm. Meanwhile, the ground operator or the automatic control program must perform dynamic mission planning according to the RUL for a subsequent period. In addition, the predictable external factors that have a significant impact on the future RUL, such as payload mass, should also be used as a basis for prediction. Based on the above requirements, the improvements of the RUL prediction algorithm required by the highly maneuverable QUAV are three-fold: (1) enhance feature expression based on multi-source sensor data flow; (2) realize real-time sequence-to-sequence prediction; (3) embed the feature with future time scale into the model.

To accomplish the above improvements, this paper provides a novel TF-based approach to predict the RUL of the highly maneuverable QUAV in real-time. The fundamental transformation of step-by-step sequential to attention-oriented parallel processing is thus complete. The TF encoder–decoder structure is used to predict the RUL sequence in the subsequent period. The feature layer fusion of the multi-sensor data reduces the dependence to a single feature and greatly improves the prediction precision and processing speed under various flight maneuvers. On the basis of the vanilla TF, the multi-scale feature mining is added to realize the distributed semantic expression of multi-source fusion with elaborate temporal characteristics. Furthermore, the external factor attention mechanism is introduced to embed external knowledge of abrupt change factors for a TF network. Consequently, the feasibility evaluation of the scheduled flight plan is provided, and the construction of the end-to-end multilevel fusion TF network model is complete. In addition, compared to the studies based on the battery discharge model, this paper considers the influence of input saturation when the highly maneuverable QUAV is flying in the boundary state [26].

To sum up, the main contributions of this paper can be summarized as follows:

1. A multi-scale feature mining process is designed for multi-sensor streaming data fusion based on the TF encoder, realizing a more effective distributed semantic expression of sensitive features.
2. An external-factor-embedding layer is constructed based on the attention mechanism, unifying the processing of features with different spatio-temporal scales.
3. An end-to-end RUL prediction method based on TF is proposed, with an accurate estimation of the RUL future sequence in real-time.

The overall organization of the paper is as follows. After a brief introduction, an overview of the proposed real-time QUAV RUL prediction method is given in Section 2. The modeling of the QUAV autopilot system, together with the simulation settings are addressed in Section 3. Section 4 details the complete TF-based RUL sequence-to-sequence prediction approach and its application. Then, in Section 5, simulation experiments with three different prediction durations are performed and analyzed. Finally, conclusions and future developments are reached in Section 6.

2. Overview of the Proposed QUAV RUL Prognostic Methodology

For the highly maneuverable QUAV, the RUL represents whether it can perform the specified maneuver missions and is determined by the cut-off voltage, as well as the maximum input throttle. The discharge voltage and throttle are affected by the flight mode (velocity), flight load, and battery condition. Unlike the full-cycle RUL of long-term and slowly degraded engines and bearings, the single-flight RUL of the highly maneuverable QUAV is no longer only related to the historical state of the aircraft, but also related to future mission maneuvers. In the field of PHM, prognostic methods are divided into one-step and sequence-to-sequence approaches. The former is to predict the current RUL value, whereas the latter is to predict the RUL value for the future time period. The current RUL will be useless if the highly maneuverable QUAV is ordered to perform a flight maneuver beyond its capability after the next sampling period. This is because the aircraft may crash immediately due to an untimely flight plan adjustment. Consequently, sequence-to-sequence prediction is the only way to solve the above issues.

In this paper, the entire RUL prediction network for the highly maneuverable QAV is established through two phases: offline training and online testing, as shown in Figure 1. The aim of the offline training phase is to build a multilevel fusion TF model that can deeply learn the battery discharge mechanism in the standard flight process. During the actual flight, mission maneuvers are typically complex, changeable, and unknown. To meet the massive data requirements of the deep learning algorithm, previous work has made efforts in either the predefined programmed maneuver or the random remote control maneuver [27]. However, the completeness still cannot be guaranteed, and much experimental cost is involved. In order to balance demand and cost, simulation training data are obtained under four standard conditions. Next, based on the simulation stop time, the RUL is calculated adaptively as the prediction label by using the linear degradation assignment method. Then, the historical dataset is generated by the sliding window interception, z-score normalization, and training–testing set split. Finally, the historical data are input into a deep learning framework to complete the offline training of the prognostic network.

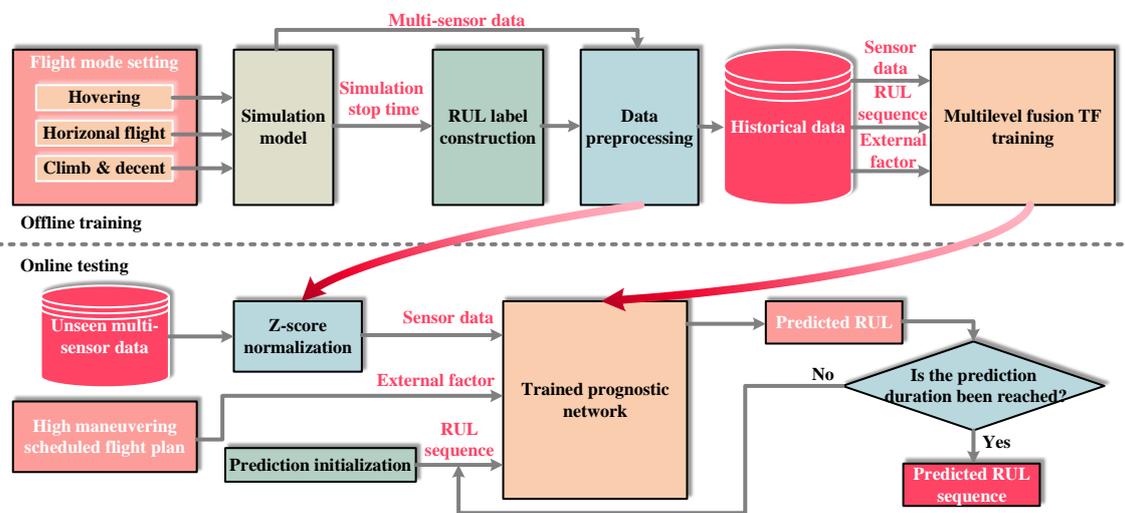


Figure 1. A graphic overview of the two-phase prognostic methodology.

In the online testing phase, unseen multi-sensor data are directly input into the trained prognostic model only after normalization. At the same time, the scheduled flight plan provides external factor data with the future time scale. In particular, after initialization, the RUL sequence is input into the decoder. The pre-trained network performs a one-step RUL prediction gradually and iteratively updates the decoder input within the expected prediction duration until the prediction is complete. By now, the prediction of the RUL sequence is complete.

It should be noted that the proposed model is run on the ground station mobile computer both for offline training and online testing. Offline data are read through the USB interface and stored on the SD card in the airborne computer. Online data are transmitted in real-time via the digital radio and Bluetooth.

3. QAV Autopilot System Modeling

The efficiency of the multi-sensor signal in the prediction of UAV flight time was verified by Sarkar et al. [14]. Additionally, not only the influence of the cutoff voltage, but also that of the input saturation on the RUL of the QAV are considered in this paper. In order to obtain the aforementioned flight data in support of the subsequent training process, a high-fidelity QAV simulation model is required. Both the predefined programmed control and random remote control can be regarded as reference control inputs, so that the QAV autopilot system simulation model is established without losing generality. As illustrated in Figure 2, the simulation model is composed of three parts: propulsion model, force and moment model, and flight control model. The modeling process of each model is detailed accordingly in a subsection below.

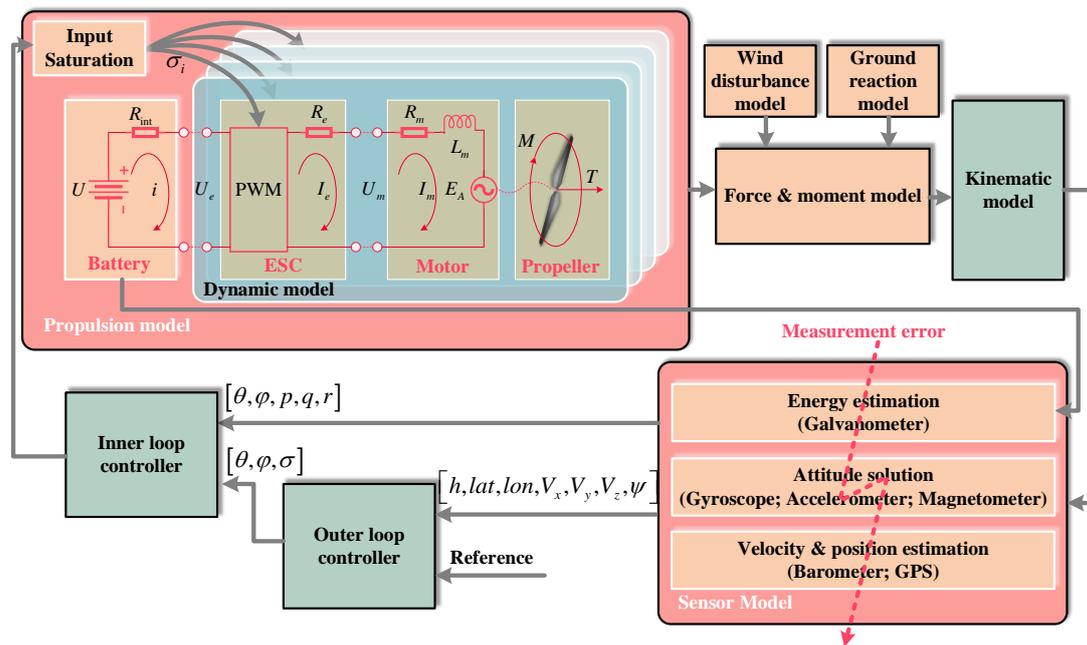


Figure 2. Structure of the simulation model for the QUAUV autopilot system.

3.1. The Propulsion Model

According to [5,28], the propulsion system is modeled. The state of charge (SOC) of the Li-Po battery is strongly related to battery voltage U , discharge current i , and power consumption P . The battery state space model is used to accurately estimate the SOC and is established as follows:

$$\begin{cases} x(k+1) = \Phi x(k) + \Gamma u(k) + w(k) \\ y(k) = Hx(k) + v(k) \end{cases}, \tag{1}$$

where $w = [\omega_1, \omega_2, \omega_3]^T$ and v are the system noise and measurement noise, respectively. $x = [R_{int}, SOC, E]^T$ is the state vector; $u = [1/E]$, $y = [U]$ is the output vector; $\Phi = I_3$, $\Gamma = [0, -P, 0]^T$, $H = [\beta_i(R_{int}), \beta_i(SOC) + \beta_{U_{oc}}(SOC), 0]$; R_{int} is the internal resistance; E represents the total energy; U_{oc} is the open circuit voltage. Suppose that a and b are independent variables and dependent variables, respectively, and $\beta_a(b)$ is the implicit expression of the functional relationships between a and b , which are defined by:

$$\begin{aligned} U_{oc} &= U_d + \lambda_1 \cdot \exp(\gamma_1 \cdot SOC) - \lambda_2 \cdot \exp(\gamma_2 \cdot SOC) \\ i &= \frac{U_{oc} - \sqrt{U_{oc}^2 - 4R_{int}P}}{2R_{int}} \end{aligned}, \tag{2}$$

where $U_d, \lambda_1, \lambda_2, \gamma_1$, and γ_2 are drone-specific parameters.

The power consumption of a QUAUV flying with the standard maneuvers is mainly determined by the flight mode and flight velocity. Hence, the QUAUV power consumption model for hovering, climbing, descending, and horizontal flight is established as follows:

$$\begin{aligned} P_h &= \frac{W^{3/2}}{\eta_h \cdot \sqrt{2\rho A_t}} \\ P_c &= \frac{W}{\eta_c(V_c)} \left(\frac{V_c}{2} + \sqrt{\frac{V_c^2}{4} + \frac{W}{2\rho A_t}} \right) \\ P_d &= \frac{W}{\eta_d(V_d)} \left(\frac{-V_d}{2} + \sqrt{\frac{V_d^2}{4} + \frac{W}{2\rho A_t}} \right) \\ P_{hor} &= \frac{W}{\eta_{hor}(V_{hor})} (V_{hor} \cdot \sin(\alpha_v(V_{hor})) + v_{hor}) \end{aligned}, \tag{3}$$

where W is the total weight of the QUAV and A_t is the total blade area. V and η are the velocities and efficiency factors under various flight modes, respectively. ρ is the air density; v_{hor} is the induced velocity during the horizontal flight; α_v is the angle of attack. The variation of η, v_{hor}, α_v with velocity can be obtained by linear fitting of the actual flight data.

It should be noted that the battery voltage is estimated with the particle filter algorithm based on (1).

After the battery model is established, the dynamic model of the actuator is carried out based on the equivalent circuit. For the QUAV, a battery powers four sets of the “electric speed controller–motor–propeller” structure. As indicated in Figure 3, the aforementioned structures have a cross-placed pattern. For simplicity, a set of dynamic structure is presented below. The electronic speed controller (ESC) converts the throttle signal, affected by the control input and battery voltage, into a pulse width modulation (PWM) signal. The motor is then driven to rotate and pushes the propeller further for the thrust force T .

The ESC working process can be described by the following formula:

$$\begin{cases} \sigma = (U_m + I_m R_e) / U_e \\ I_e = \sigma I_m \\ U_e = U - i R_{int} \end{cases}, \tag{4}$$

where R_e is the resistance of the ESC. σ is the throttle input. U_e and I_e are the input voltage and input current, respectively. U_m and I_m represent the equivalent voltage and the equivalent current, respectively.



Figure 3. A real drone referenced in the paper.

Equation (4) is followed by the motor model:

$$\begin{cases} U_m = f_{U_m}(\Theta_m, M, N) \\ I_m = f_{I_m}(\Theta_m, M, N) \end{cases}, \tag{5}$$

where M is the load torque and N is the motor speed. $\Theta_m \triangleq \{K_V, I_{m0}, R_m\}$ is the motor-specific parameter; K_V represents the KV value, I_{m0} is the no-load current; R_m is the resistance of the motor.

Then, the aerodynamic model of the propeller installed on the motor is established as follows:

$$\begin{cases} T = C_T \rho \left(\frac{N}{60}\right) D_p^4 \\ M = C_M \rho \left(\frac{N}{60}\right) D_p^5 \end{cases}, \tag{6}$$

where N is the propeller speed, C_t is the propeller thrust coefficient, C_m is the propeller torque coefficient, and D_p is the blade diameter.

3.2. The Force and Moment Model

According to [29], environmental influence has to be taken into account in modeling. In this paper, a constant wind pattern is developed to simulate wind field interference:

$$V_w \propto T. \tag{7}$$

Taking into account the ground reaction force F_c , the model of the total force acting on the QUAUV is given as follows:

$$\begin{cases} F = F_a + F_g + F_c + T \\ F_a = f_{F_a}(V_w, V) \\ F_c = f_{F_c}(h_{env}, h) \end{cases}, \tag{8}$$

where F_a and F_g are the aerodynamic force and the force of gravity, respectively. h_{env} represents the ground height.

3.3. Flight Control Model

The QUAUV flight controller has three main functions: altitude control, velocity control, and attitude control. Before building the control model, it is necessary to describe the state of the drone with the kinematic model. The state measurement is then simulated by the sensor model to perform the closed-loop feedback. The dynamic model is represented by a set of six-degree-of-freedom 13-state high-fidelity nonlinear equations:

$$\begin{cases} \dot{V}_x = \frac{T}{m}(\sin \theta \cos \psi \cos \phi + \sin \psi \sin \phi) - \frac{D_x V_x^2}{m} \\ \dot{V}_y = \frac{T}{m}(\sin \theta \cos \psi \sin \phi - \cos \psi \sin \phi) - \frac{D_y V_y^2}{m} \\ \dot{V}_z = \frac{T}{m} \cos \theta \cos \phi - g - \frac{D_z V_z^2}{m} \\ \ddot{\phi} = \frac{J_y - J_z}{J_x} \dot{\theta} \dot{\psi} + \frac{j_r \dot{\theta}}{J_x} (-N_1 + N_2 - N_3 + N_4) + \frac{U_2}{J_x} \\ \ddot{\theta} = \frac{J_z - J_x}{J_y} \dot{\phi} \dot{\psi} + \frac{j_r \dot{\phi}}{J_y} (-N_1 + N_2 - N_3 + N_4) + \frac{U_3}{J_y} \\ \ddot{\psi} = \frac{J_x - J_y}{J_z} \dot{\phi} \dot{\theta} + \frac{U_4}{J_z} \end{cases}, \tag{9}$$

where V_x , V_y , and V_z represent the three-axis velocity in the Earth coordinate system, D_x , D_y , and D_z are the air-drag coefficients in the three-axis direction, m is the total mass, g indicates the gravitational acceleration, $J = \text{diag}(J_x, J_y, J_z)$ is the inertia tensor, and j_r is the propeller moment of inertia. U_2 , U_3 , and U_4 denote the roll, pitch, and yaw moment of the body, respectively.

The sensor observation is not only the basis of the autopilot control, but also provides data support for the subsequent design of the prognostic algorithm. In this paper, the battery voltage and discharge current are obtained by a power module to estimate the energy condition and are integrated for the SOC. The flight velocity and position are obtained from a GPS and a barometer, respectively. The flight attitude is derived from a gyroscope, accelerometer, and magnetometer. For a sensor, the measured signal can be obtained by adding a particular Gaussian noise interference term:

$$z(t) = y_d(t) + g(t), \tag{10}$$

where y_d is the system output, $g = k_g \cdot y_d$ represents the measurement error, and k_g is the noise gain.

As the simple and effective PID control method is adopted, the control parameters are tuned from the inner loop to the outer loop in the design process.

3.4. Simulation Setup and Output

The autopilot model established by (1) shows that the battery voltage, discharge current, SOC, load weight, and flight velocity are strongly related to the RUL. Moreover, the throttle has a nonlinear relationship with the RUL due to the influence of input saturation. Factor selection in Ref. [14] needs to rely on the manual correlation analysis, while the deep learning methods achieve an end-to-end classification that can significantly reduce labor and time costs. The effect of predicting the RUL sequence for the highly maneuverable QUAV is affected by the following two complex factors: (1) multi-sensor historical observation signals; (2) external factors for a future duration. The former includes the battery voltage, the discharge current, the throttle, and the SOC. Among them, the SOC is obtained indirectly by integrating the discharge current value, and the throttle is calculated as the efficiency ratio of the battery $\sigma = U/U_b$, where U_b is the nominal voltage. The latter includes the total weight and the flight velocity, which are taken from the reference control signal to simulate the scheduled flight plan for the actual flight.

The constructed high-fidelity simulation model is run to fetch the data above. Assuming that there is a power loss at the beginning of each flight, the initial SOC is reduced slightly at random as each simulation starts.

Definition 1. The flight purpose of the highly maneuverable QUAV studied in this paper is to execute the mission maneuvers. The maximum RUL is the time from the start of the flight until the drone cannot execute a specific instruction and is unable to maintain the hover.

Full-cycle flight data are the output sequence obtained between the start and the automatic end of a simulation. It is considered that the RUL is reduced to 0 at the end of the simulation. The simulation stops and the stop time is recorded as the end of life when one of the following conditions is met: (1) the battery voltage drops to the cut-off voltage; (2) the hover throttle exceeds the maximum throttle constrained by the minimum attitude control capability.

4. The Proposed Method

Figure 4 provides a framework for the approach proposed in this paper. Based on the vanilla TF offered by Google [30], we made necessary improvements to the encoder and decoder, respectively, for the needs of the highly maneuverable QUAV. Specifically, our multilevel fusion TF is mainly composed of the following two major parts: (1) multi-scale multi-sensor fusion attention; (2) external factors' influence.

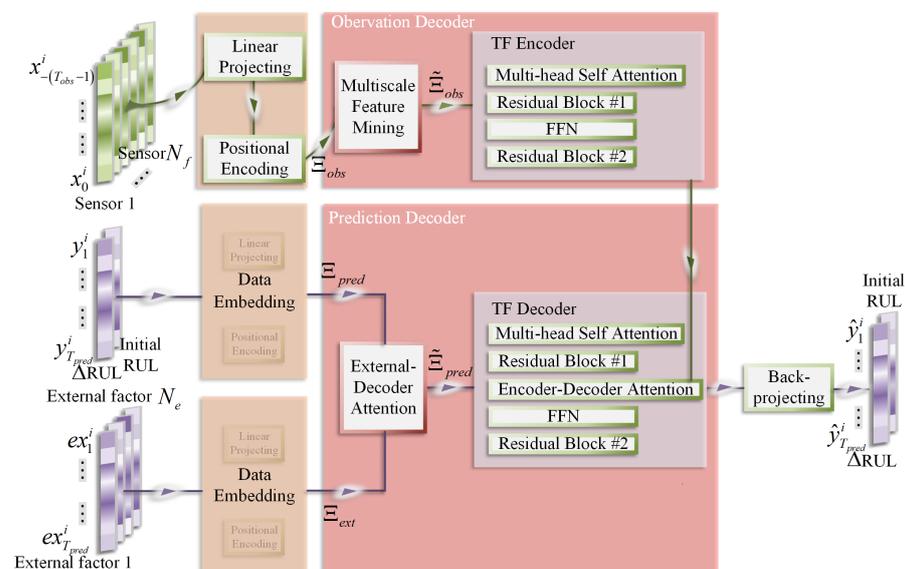


Figure 4. Model illustration of the proposed multilevel fusion transformer network.

4.1. Data Preprocessing and Notation Statement

Once the simulation data are obtained, the RUL target label must be constructed. The RUL curve of the QUAV varies with the change of the simulation condition.

Remark 1. *The initial value of the RUL sequence (first hitting time (FHT)) is the simulation time, which is the longest time the drone can perform the current maneuver.*

The influence of system degradation on a one-time flight is very slight and can be ignored. The RUL data are generated adaptively using the linear descent assignment method as the prediction label:

$$RUL(t) = \min(\{t|\sigma_{hover}(t) \geq \sigma_{max}\} \cup \{t|U(t) \leq U_{thr}\}) - t, \tag{11}$$

where σ_{hover} is the hovering throttle, σ_{max} is the maximum throttle limit (input saturation), and U_{thr} is the cut-off voltage.

For the i -th flight, the observations $\Gamma_{obs} = \{\mathbf{x}_t^{(i)}\}_{t=-(T_{obs}-1)}^0, \mathbf{x}_t^{(i)} = (x_t^{i,1}, x_t^{i,2}, \dots, x_t^{i,N_f}) \in \mathbb{R}^{N_f}$ composed of multi-sensor signals are constructed by intercepting the acquired data at interval T_{slid} . The predictions $\Gamma_{pred} = \{\mathbf{y}_t^{(i)}\}_{t=1}^{T_{pred}}, \mathbf{y}_t^{(i)} = (\Delta RUL_t^i, InitRUL_t^i) \in \mathbb{R}^2$ are composed of the variation of the RUL and the initial value when prediction starts. The external inputs $\Gamma_{ext} = \{\mathbf{ex}_t^{(i)}\}_{t=1}^{T_{pred}}, \mathbf{ex}_t^{(i)} = (ex_t^{i,1}, ex_t^{i,2}, \dots, ex_t^{i,N_e}) \in \mathbb{R}^{N_e}$ are composed of the external factor signals. The size of each dataset is $\lfloor RUL(0) / (T_{obs} + T_{pred} + T_{slid}) \rfloor + 1$. After the z-score standardization, data are divided into the training set and the testing set according to a certain proportion. It should be noted that the observations, predictions, and external inputs form the historical dataset.

4.2. Data Embedding

The purpose of data embedding is twofold: (1) to realize the multi-sensor data fusion; (2) to add temporal information to parallel processing. The former is realized by linear projection, and the latter is realized by “positional encoding”, both of which are indispensable.

The original input is mapped to the high-dimensional feature space by linear projection $f_{linear} \rightarrow \mathbb{R}^D$ to realize the distributed expression, allowing the multilevel fusion TF to process the input features:

$$\tilde{\mathbf{x}}_{obs}^{(i,t)} = f_{linear}(\mathbf{x}_t^{(i)}) = \mathbf{x}_t^{(i)T} W_x, \tag{12}$$

where W_x is a coefficient matrix. The high-dimensional space projection operation not only refines the spatial features of the input data, but also performs the feature layer fusion of sensor signals and provides an input interface for TF. Although advanced signal processing technology can map high-dimensional features, it has to perform complex time–frequency domain feature calculation and sensitive feature selection manually. In contrast, linear projection can complete both tasks at the same time by simply training the network, which is of great significance.

Correspondingly, the output of the i -th flight at time t is a vector with D dimensions. Outputs will be projected back into the prediction space through the inverse transformation of (12), so as to realize the embedding of the predictions.

TF realizes the feature layer fusion of multi-sensor data by taking into account the correlation of data in different spatial features at different times based on the attention mechanism. TF can embed data both without and with “position encoding”. However, the input is considered to be a set of vectors without sequential order in the attention layers. Therefore, the attention layers are insensitive to temporal information. To address this issue, “positional encoding” is added to ensure the temporal uniqueness of data. This

operation is applied to encode each historical and future time, and a corresponding time stamp $p^t = \{p_{t,d}\}_{d=1}^D$ is added for each input to be embedded:

$$\zeta_{obs}^{(i,t)} = p^t + \tilde{x}_{obs}^{(i,t)}, \tag{13}$$

where the time stamp is defined as $\begin{cases} p_{t,2k} = \sin(t \cdot 10000^{-2k/D}) \\ p_{t,2k-1} = \cos(t \cdot 10000^{-(2k-1)/D}) \end{cases}, k = 1, 2, \dots, d/2$ to keep the value unique in 10,000 time steps.

4.3. The Multilevel Fusion Transformer Network Model

To solve the problems where the vanilla TF is not very sensitive to temporal features and information in the future time period is ignored, a TF-based method is proposed in this paper. It is improved by the multilevel fusion operation, which is achieved by extracting multi-scale temporal features and merging features with different durations.

4.3.1. The Multi-Scale Spatiotemporal Feature Mining

When the highly maneuverable QUAV carries a load with the weight limit or flies at an extreme speed, it tends to lose control during takeoff. These are due to the input saturation or the reduction of the driving voltage to the cut-off voltage level, resulting in an accident. In order to shorten the blank period of the prediction at the beginning, TF is expected to use as few observations as possible to achieve the expected effect. In this case, the temporal information of the observation is limited. Inspired by the idea of multi-grained scanning in deep forest [31], a multi-scale feature mining mechanism is developed in this paper. Combined with the TF encoder, the temporal information in the embedded multi-sensor data is deeply mined to find the spatiotemporal channel.

Once the multi-sensor input $\Xi_{obs} = \{\zeta_{obs}^{(i)}\}_{t=-(T_{obs}-1)}^0 \in \mathbb{R}^{T \times D}$ is obtained according to Section 4.1, the multilevel fusion TF performs the 1D convolution on the input tensor along the spatiotemporal dimension using the mining operator $M(\cdot; \Theta)$. Assigning the kernel size k and the padding size p , $\Theta := (k, p)$ determines the mining scale. For the specific s scales, the mining operator processes the input tensor on each scale to obtain the tensor with the spatiotemporal dimension as $\tilde{T} = T + 2p - k + 1$. D mining operators are placed simultaneously in each scale to keep the dimension of the multi-sensor feature. After the spatiotemporal features are multi-scale refined, the elite features are selected as the processing results:

$$\tilde{\Xi}_{obs} = \max\left(\{M(\Xi; \Theta_k)\}_{k=1}^{2s-1}\right), s = 1, 2, \dots \tag{14}$$

It should be noted that the above formula restricts the corresponding relation between k and p : $p \in \{p | \tilde{T} = T, k\}$.

4.3.2. Construction of the Transformer Network Encoder–Decoder Model

The integrated TF encoder and decoder are composed of multiple basic layers with the attention mechanism. Each basic layer has three components: multi-head self-attention module, feed-forward module, and two residual connection modules.

The multi-head self-attention module is realized by parallel calculation of h self-attention modules. For j self-attention modules, the trainable hyperparameters: query tensor $Q_j = \tilde{\zeta} W_j^Q$, key tensor $K_j = \tilde{\zeta} W_j^K$, and value tensor $V_j = \tilde{\zeta} W_j^V$ are determined by the query matrix W_j^Q , key point matrix W_j^K , and value matrix W_j^V , respectively. Together, they form the attention-based weight calculation mechanism:

$$Attention(Q_j, K_j, V_j) = softmax\left(\frac{Q_j K_j^T}{\sqrt{d_k}}\right) V_j, \tag{15}$$

where $d_k = D/h$ is the dimension of the matrix made up of hyperparameters.

After each self-attention module is calculated, the parallel attention calculation is applied to realize the integration of information from different representation subspaces:

$$MultiHead(Q, K, V) = Concat\left(\{Attention(Q_j, K_j, V_j)\}_{j=1}^h\right)W^A, \tag{16}$$

where W^A is the matrix of attention and $Concat(\cdot)$ represents the tensor concatenation.

The feed-forward module is composed of a linear transformation and the ReLU activation function, which acts on each observation time step with the same weight:

$$FFN(\xi_{obs}^{(i,t)}) = ReLU\left(\xi_{obs}^{(i,t)T}W_{\xi_{obs},1} + B_{\xi_{obs},1}\right)^T W_{\xi_{obs},2} + B_{\xi_{obs},2}, \tag{17}$$

where $W_{\xi_{obs}}$ and $B_{\xi_{obs}}$ are coefficient matrices.

4.3.3. External Factor Fusion

According to the state space model (1), the power consumption model (3), and the simulation stop conditions, the RUL is affected by the load weight and flight velocity. These external factors are abrupt signals, and their impact is unpredictable without symptoms. This cannot match the prognostic demand for the highly maneuverable future flight capability. To make TF consider the strong correlation between the RUL sequences and external factors, the external factor fusion is realized by the external decoder attention layer added to the decoder. The purpose of the encoder in Section 4.3.2 is to create a spatiotemporal sequence representation for embedded multi-sensor signals, so as to grant the TF network memory. Simultaneously, its key tensor K_{enc} and value tensor V_{enc} will be shared with the decoder. External factors can be regarded as similar to multi-sensor observation features, but located in a different temporal space. Predictability makes them work directly over the prediction duration. Therefore, the operation in Section 4.2 is applied to the input external factor tensors Γ_{ext} for calculating $\Xi_{ext} = Embed(\Gamma_{ext}) \in \mathbb{R}^{T_{pred} \times D}$. In order to prevent the predictable information from changing the attention to historical observations, features are coupled and updated in the decoder embedding stage rather than in the encoder–decoder attention stage:

$$\tilde{\Xi}_{pred} = MultiHead(\Xi_{ext}, \Xi_{ext}, \Xi_{pred}). \tag{18}$$

Through the external factor fusion, the learned potential representations are transmitted to the TF, which reinforces the importance of external factors and the their network attention.

4.4. Offline Training and Online Predicting

The proposed prognosis model built in offline training learns the nonlinear functions of the RUL with the Li-Po battery discharge failure, external factors, and multi-sensor data. In the training process, the Adam optimizer is used for back-propagating to minimize loss and achieve nonlinear fitting. The loss function is defined as follows:

$$\ell(\theta) = PairwiseDistance(\hat{Y}_{pred}^i, Y_{pred}^i), \tag{19}$$

where θ represents the trainable hyperparameter. $\hat{Y}_{pred}^i \in \mathbb{R}^2$ is the prediction output, and $PairwiseDistance(X, Y)$ represents the Euclidean distance at the pixel level. The trained model will be directly applied to perform the online prediction, as detailed in Algorithm 1. Precise prediction of the RUL can be accomplished in various complex flight processes of the highly maneuverable QUAV.

Algorithm 1 The online predicting process based on the multilevel fusion transformer network.

Input: Historical dataset, observation time T_{obs} , prediction time T_{pred} , multi-sensor observations $\{\mathbf{x}_t\}_{t=-(T_{obs}-1)}^0$, external factors $\{\mathbf{ex}_t\}_{t=1}^{T_{pred}}$.

Output: RUL prediction $\{RUL_t\}_{t=1}^{T_{pred}}$
 Randomly initialize the multilevel fusion TF with hyperparameter θ .
 Train the prognostic network based on (19).

Initialization: the decoder input $\{\mathbf{y}_t\}_{t=0}^1 = [0, 0]$.

- 1: **for** $n_{pred} = 1$ to T_{pred} **do**
- 2: **for** $n = 1$ to n_{pred} **do**
- 3: Initialize the decoder mask tensor $M_{enc} = ones(n_{pred}, n_{pred})$ (composed of "0" and "1": "0" representing "mask" and vice versa);
- 4: Mask: mask out the elements in the decoder input who ranked after n in the temporal dimension ($M_{enc}(n, (n + 1) : n_{pred}) := 0$);
- 5: **end for**
- 6: Predict: calculate predictions $\{\hat{\mathbf{y}}_t\}_{t=1}^{n_{pred}}$ according to (15)–(18);
- 7: Update: concatenate to update the decoder input $\{\mathbf{y}_t\}_{t=0}^{n_{pred}} = [\{\mathbf{y}_t\}_{t=0}^{n_{pred}}; \hat{\mathbf{y}}_{n_{pred}}]$
- 8: **end for**
- 9: Calculate RUL according to $\{\widehat{RUL}_t\}_{t=1}^{T_{pred}} = \left\{ \sum_{n=1}^{n_{pred}} \mathbf{y}_t(1) \right\}_{n_{pred}=1}^{T_{pred}} + \{\mathbf{y}_t(2)\}_{t=1}^{T_{pred}}$

5. Simulation and Result Analysis

At the beginning of this section, the implementation details are provided as follows to obtain the simulation results: the PC is configured with a GeForce RTX 3060 GPU and an Intel Core i7 CPU; the autopilot model is built within the Simulink environment of MATLAB R2020b; the prediction algorithm is programmed based on PyTorch 1.10.

5.1. Performance Metrics

In order to make the performance of the RUL prediction algorithm more intuitive, five metrics are employed in this paper. Among them, the mean absolute error (MAE), the mean-squared error (MSE), the mean absolute percentage error (MAPE), and the cumulative relative accuracy (CRA) are widely used, as in [32] and other works, while the mean percentage error (MPE) is chosen according to the characteristics of the QUAV RUL prediction [33]. These metrics are defined as follows:

$$\begin{aligned}
 MAE &= \frac{\sum_{n=1}^N |h(t_n)|}{N} \\
 MSE &= \frac{\sum_{n=1}^N (h(t_n))^2}{N} \\
 MAPE &= \frac{100\%}{N} \sum_{n=1}^N \left| \frac{h(t_n)}{\{RUL_t\}_{t=t_n}^{t_n+T_{pred}-1}} \right|, \\
 CRA &= \frac{1}{N} \sum_{n=1}^N \left(1 - \left| \frac{h(t_n)}{\{RUL_t\}_{t=t_n}^{t_n+T_{pred}-1}} \right| \right) \\
 MPE &= \frac{100\%}{N} \sum_{n=1}^N \left(\frac{h(t_n)}{\{RUL_t\}_{t=t_n}^{t_n+T_{pred}-1}} \right)
 \end{aligned} \tag{20}$$

where N is the total number of predictions within an entire flight, $h(t_n) = \{\widehat{RUL}_t\}_{t=t_n}^{t_n+T_{pred}-1} - \{RUL_t\}_{t=t_n}^{t_n+T_{pred}-1}$, and t_n represents the start time of the n -th prediction.

In addition, the real-time performance of the prediction methods is constrained by the following conditions: the algorithm processing time must be less than the sampling time.

Therefore, this paper takes the algorithm processing time as the standard to measure the real-time performance of the method and as one of the performance metrics.

5.2. Simulation Dataset Generation

Simulation parameters are fit through actual flight experiments with multiple flight modes, variable flight velocities, and variable load masses. Based on the above, the autopilot model mentioned in Section 2 is constructed. The real drone used in the experiment is depicted in Figure 3, and its configurations are presented in Table 1. Except for the minimal set of sensors incorporated into Pixhawk Series flight controllers, the real drone is equipped with a GPS and a power module. The sensor specification is presented in Table 2. The GPS and power module are connected to the Pixhawk 2.4.8 through serial ports for data acquisition assurance. The PID controller is used to perform automatic control during simulated flight. The flight control model parameters are listed in Table 3.

Table 1. Configurations and parameters of the real drone.

Item	Symbol	Value	Units
Frame diagonal length	D_f	550	mm
Empty mass	m_0	1.357	kg
Gravitational acceleration	g	9.8	N/kg
Propeller (T-MOTOR T9545-B)	Blade diameter	D_p	0.241
	Total blade area	A_t	0.183
	Propeller thrust coefficient	C_t	1.016×10^{-5}
	Propeller torque coefficient	C_m	1.392×10^{-7}
Motor (T-MOTOR Air Gear 350 KV920)	KV value	K_v	920
	No-load current	I_{m_0}	0.5
	Motor resistance	R_m	132
ESC (T-MOTOR AIR 20A)	ESC resistance	R_e	8
Battery (5100mAh 4S Li-Po)	Nominal voltage	U_b	12.6
	Initial internal resistance	$R_{int}(0)$	27
	Initial total energy	$E(0)$	2.024×10^5
	Specific parameters	$[U_d, \lambda_1, \lambda_2, \gamma_1, \gamma_2]$	$[11.731, 0.058, 2.524, 3.489, -7.932]$
	Air density	ρ	1.15
	System noise	w	$[3.46 \times 10^5, 3.41 \times 10^5, 13.28]^T$
	Measurement noise	v	0.33

Table 2. Sensor specification.

Part	Package Size (mm ³)	Sampling Rate (Hz)	Power Consumption (mW)	Weight (g)	Accuracy	Serial Interfaces Supported
M8N	50 × 50 × 12.8	10	50	30	0.1 m/s	USB, SPI, UART
MS5611	5 × 3 × 1	2000	3.3×10^{-3}	0.10	1.5 mbar	I ² C, SPI
L3GD20	4 × 4 × 1	95	18.3	0.10	8.75 mdps	I ² C, SPI
MPU6000	4 × 4 × 0.9	8	13.2	0.14	2–3%	I ² C, SPI
PM02D	25 × 21 × 9	10	5	20	1%	I ² C

Table 3. Flight control model parameters.

Item	Symbol	Value	Units
Inertia tensor	J	$\begin{bmatrix} 0.020 & 0 & 0 \\ 0 & 0.020 & 0 \\ 0 & 0 & 0.036 \end{bmatrix}$	$\text{kg} \cdot \text{m}^2$
Propeller moment of inertia	j_r	1.03×10^{-4}	$\text{kg} \cdot \text{m}^2$
Sensor noise gain	GPS (M8N)	$k_{g,GPS}$	0.200
	Barometer (MS5611)	$k_{g,baro}$	0.025
	Gyroscope (L3GD20)	$k_{g,gyro}$	0.025
	Accelerometer (MPU6000)	$k_{g,acc}$	0.030
	Magnetometer (MPU6000)	$k_{g,mag}$	0.050
	Power module (PM02D)	$k_{g,gal}$	0.054

The flight mode of the standard conditions is set according to the following rules: the load mass of the drone is less than 0.5 kg; the flight space volume is $1000 \times 1000 \times 200 \text{ m}^3$; the horizontal flight velocity is $\{V_{hor,x}, V_{hor,y}\} \in [-8, 8] \text{ m/s}$; the maximum descent velocity is $V_{d,max} = 2 \text{ m/s}$; the maximum climb velocity is $V_{c,max} = 4.5 \text{ m/s}$; the maximum throttle is 0.95; the cut-off voltage is 10.3 V. According to battery voltage variation characteristics and control requirements, the simulation experiments with a sampling time of 1s provide 850 sets of flight data under four standard conditions under the influence of different external factors mentioned above. For a certain simulation, the drone will immediately perform a hover, climb, or descent, depending on the set flight mode, after taking off as fast as possible. In horizontal flight conditions, the drone flies in a straight line to the edge of space and hovers thereafter. As soon as it reaches the upper limit of the space height under climb flight conditions, the drone will immediately descend. The flight mode described above is visualized in Figure 5, which demonstrates that the offline training dataset required in this paper is easy to obtain. Because there is no need for massive and disordered experiments, the cost of labor, economy, and time are significantly reduced. However, the small number of samples also puts forward higher requirements for the feature extracting and learning ability of the deep learning model.

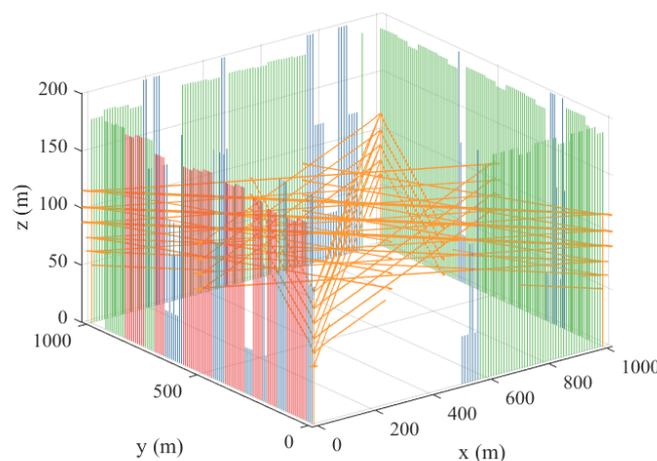


Figure 5. The QUAV trajectories of the flight mode (under the standard conditions) for generating the historical data. The blue and yellow lines represent the hover and horizontal flight, respectively. The red lines represent the climb flight, and the green lines represent that the QUAV climbs for a while and then descends.

Previous studies generally adopted the battery voltage as a single feature and considered the RUL of the UAV to be 0 at the end of discharge (EOD). However, for the UAV

flying in a high maneuver, this judgment condition is one-sided. Since there are usually sudden and drastic maneuver changes, the drone is subjected to the current maximum throttle that the battery can provide. Therefore, input saturation is taken as one of the termination conditions for the drone’s useful life, for the drone cannot perform the flight plan as scheduled in this case or is even out of control and crashes. As shown in Figure 6a, the time when the RUL of the highly maneuverable UAV is 0 may be earlier than the time of the EOD. Hence, it is not feasible to predict only based on the battery voltage, so the multi-sensor feature is needed. The changes in the discharge current and throttle with time are also depicted in Figure 6b,c, respectively, indicating that multi-sensor features have different degrees of significant influence on the RUL. In addition, the nonlinear mapping relationship between features and the RUL is very complex, which also makes the application of the deep learning approach necessary.

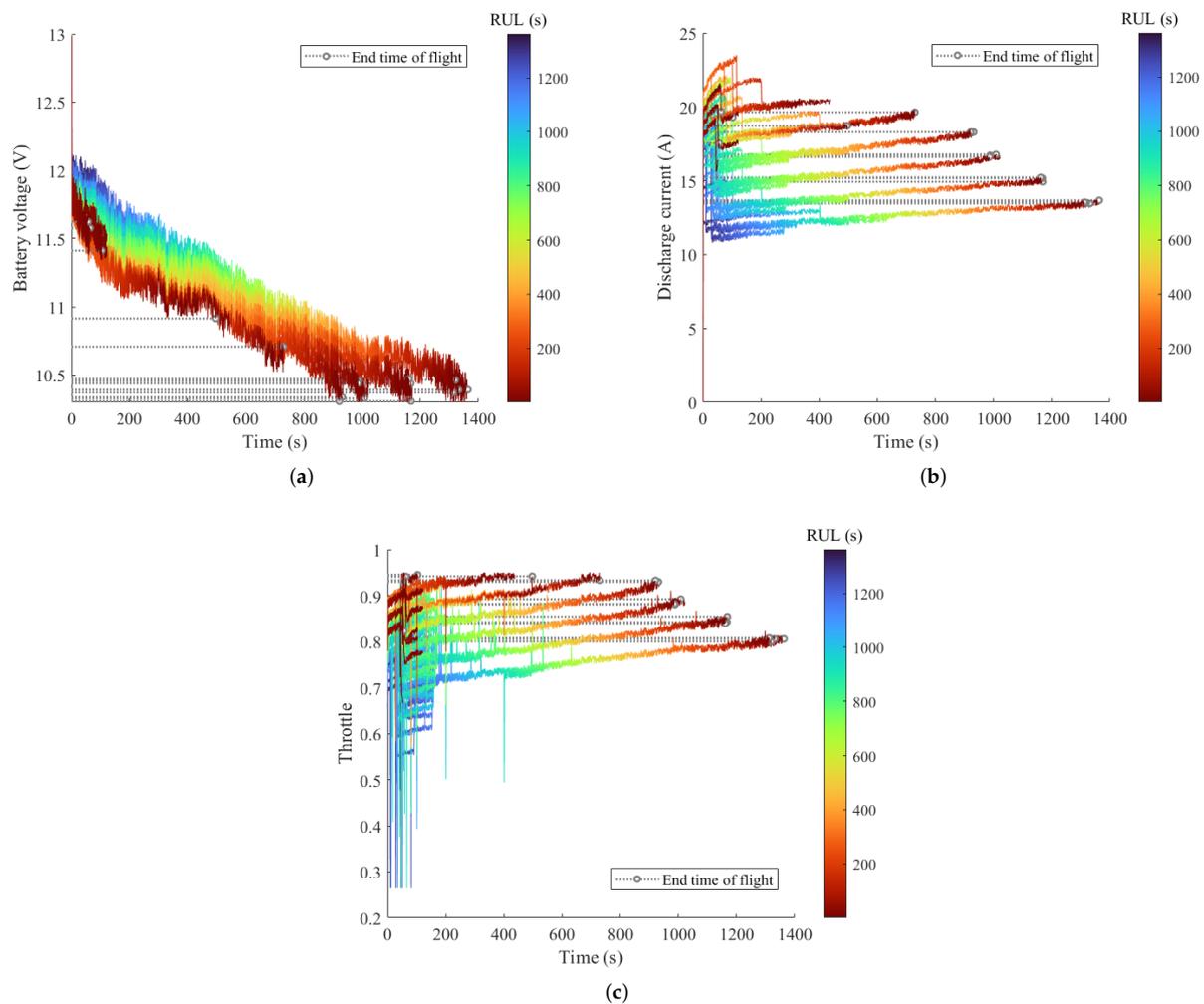


Figure 6. Multi-sensor signal in the historical flight data of the highly maneuverable QUAV. (a) Battery voltage. (b) Discharge current. (c) Throttle.

As the simulation data indicate, the shortest operating time of the drone in extreme conditions is about 49 s, and the longest time between takeoff and steady flight is approximately 30 s. Hence, based on the requirements of the margin and blank period, the observation time of 32 s was selected. After the flight data are processed by the sliding window with both a width and sliding distance of $(32 + T_{pred})$ s, the intercepted data will form the historical dataset. The multi-sensor observations dimension is 32×7 ; the external factor input dimension is $(32 + T_{pred}) \times 4$; the predictions’ dimension is $(32 + T_{pred}) \times 2$.

The multilevel fusion TF parameters are defined as follows: the embedding size $D = 512$; the multi-scale mining kernel has three scales $k = 1, 2, 3$; the corresponding

padding size is $p = 0, 0, 1$. The encoder and decoder are composed of six basic layers, and eight attention heads are included in each multi-head self-attention module. During model training, the batch size is 80, the maximum iteration is 30, and the ratio of training to testing set size is 3:1.

By randomly selecting a sufficient number of test sets in the dataset, we externally verified and analyzed the results of different prediction methods for different purposes. The comparative results are shown in Table 4. The results show that the training and testing loss of the multilevel fusion TF converges to a lower level. The proposed method has better generalization ability.

Table 4. External validation result.

Method	One-Step Prediction		Sequence-to-Sequence Prediction			
	Train Loss ¹	Test Loss ²	48 s		128 s	
			Train Loss ¹	Test Loss ²	Train Loss ¹	Test Loss ²
Multilevel Fusion TF	0.0398	14.9920	0.0553	24.6824	0.1034	47.3783
TF	0.0736	18.5021	0.0998	29.4287	0.1688	53.1905
Vanilla TF	-	-	0.1188	32.0245	-	-
LSTM	0.1144	23.5549	-	-	-	-

¹ Pairwise distance. ² Mean Euclidean distance (s^2).

To prove that the trained prediction model can achieve the real-time, high-precision RUL prediction for the highly maneuverable QUAV, a representative flight plan was designed to evaluate the prediction performance. Subsequent sections will provide a detailed analysis of the prediction results based on the plan shown in Figure 7.

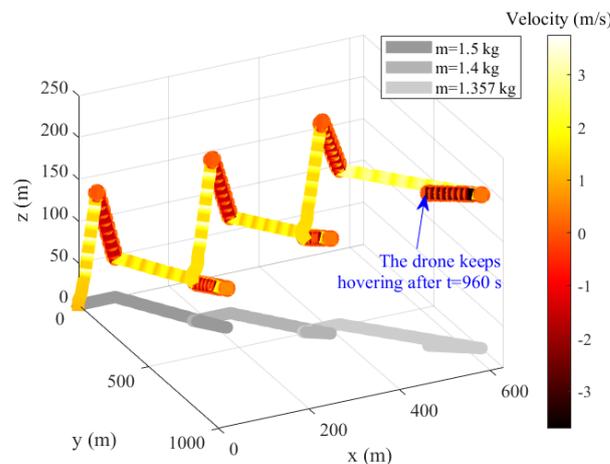


Figure 7. The representative highly maneuverable flight plan. It should be noted that the QUAV climbs and descends vertically, that is the horizontal displacement is depicted to show the motion state more intuitively, but does not really exist.

Since the ground station mobile computer is resource constrained, the computational complexity of data-driven methods should be considered. The computational complexity of the proposed algorithm and the mainstream advanced machine learning algorithm are calculated, respectively, and the comparative results are given as shown in Table 5. It can be seen that the parallel processing mechanism of TF significantly reduces the time complexity of each layer, and it is more suitable for processing sequence data. Although the TF-based method has a slightly larger space complexity for parallel computing, it only occupies 225.144 MB of memory under the above setting conditions, which fully meets the storage requirements.

Table 5. Per-layer computational complexity and minimum number of sequential operations for different layer types. $n = T_{obs}$ is the sequence length; d is the node dimension; k_1 and k_2 are kernel sizes ($k_1 > k_2$); s is the random permutations ($s \gg n$).

Layer Type	Computational Complexity Per Layer		Sequential Operations
	Time Complexity	Space Complexity	
Multilevel Fusion TF	$O((k_1 + n) \cdot n \cdot d)$	$O(n^2)$	$O(1)$
Vanilla TF	$O((k_2 + n) \cdot n \cdot d)$	$O(n^2)$	$O(1)$
LSTM	$O(n \cdot d^2)$	$O(n)$	$O(n)$
GBT	$O(s \cdot d^2)$	$O(s)$	$O(n)$

5.3. One-Step RUL Prediction Result

As discussed in Section 1, LSTM and other machine learning methods are unable to reach sequence-to-sequence prediction on their own. Therefore, in order to prove the superiority of the proposed method in the one-step prediction, the corresponding simulation experiments are first performed. Among them, the TF model (without the external factor fusion) parameters are the same as above; the LSTM consists of four hidden layers with 512 state nodes per layer. The one-step prediction results of the above methods are shown in Figure 8a. In addition, as a representative of the machine learning method, gradient boosted trees (GBTs) proved to have a good performance in predicting the RUL of the QUAV. Hence, the results from [25] are cited for comparison. Detailed performance metrics and visualizations are illustrated in Table 6 and Figure 8b, respectively.

Table 6. One-step RUL prediction performance.

Metric	Multilevel Fusion TF	TF	LSTM	GBT
MSE (s^2)	3.6863×10^3	4.3887×10^3	8.1377×10^3	8.8324×10^3
MAPE (%)	24.1185	28.3274	48.7357	52.2349
CRA	0.7588	0.7233	0.5126	0.4776
MAE (s)	75.2344	80.8965	102.1218	110.6437
Time (ms)	0.212	0.212	0.125	0.185

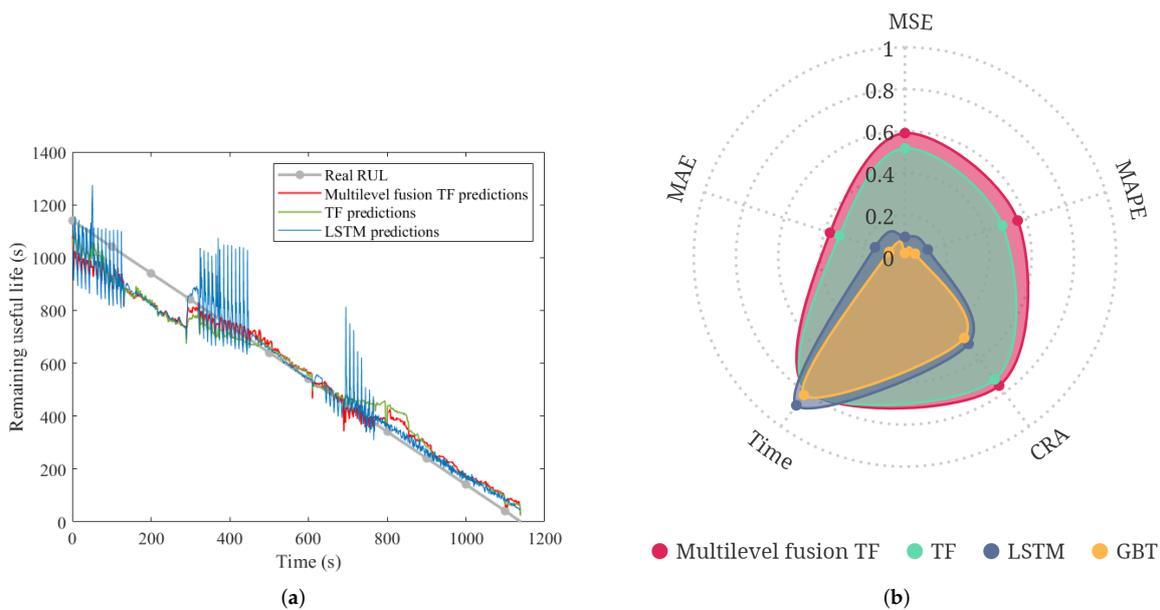


Figure 8. The one-step prediction results. (a) Comparative results of RUL predictions. (b) The 5-dimensional radar chart for the performance comparison of prediction methods. The performance depicted in the radar chart ranges from 0 to 1 (from low to high).

It can be seen that the multilevel fusion TF sacrifices some rapidity, but greatly improves the prognostic precision. This indicates that the proposed method has significant advantages in learning the multi-sensor signal features. Figure 8 and Table 6 show that both the LSTM and GBT accuracies are unsatisfactory, although their processing speed slightly improved. However, the processing time of the proposed method in this paper is still far less than the sampling time of 1 s, which will not cause data stack and time delay accumulation. Therefore, the real-time requirement is fully met. Figure 8a shows that LSTM has a poorer RUL prediction performance in the early stage of the flight with frequent change maneuvers compared to that in the post-hover stage (after 960 s). This is because it focuses only on the sequential features of time rather than spatiotemporal features, so it is continually affected by the abrupt changing throttle signal throughout the prediction process. The self-attention module allows the TF to change its attention to different signals at different times and makes the end-to-end learning possible.

5.4. Sequence-to-Sequence RUL Prediction Result

The multi-scale feature mining and external factor fusion mechanisms designed in this paper are for the improvement of the vanilla TF. To demonstrate their feasibility and superiority, the RUL prediction experiment for the next 48s was carried out. The prediction results of the above methods are presented in Figure 9a–c, and the detailed performance metrics and visualization are provided in Table 7 and Figure 9d, respectively.

Table 7. Performance of the RUL prediction for the next 48 s.

Metric	Multilevel Fusion TF	TF	Vanilla TF
MSE (s ²)	6.3354×10^3	2.8332×10^3	4.8163×10^3
MPE (%)	1.3656	8.2388	7.6658
CRA	0.8678	0.8524	0.8371
MAE (s)	58.4723	61.3311	66.8219
Time (ms)	17.341	16.409	15.259

For the RUL, if the predicted value is greater than the real value, this will lead to an overestimation of the QUAV remaining flight capability. That is, when the relative errors are equal, we prefer the predicted value to be lower than the real one. Therefore, the MPE is applied, and the smaller the value, the better. Table 7 shows that the TF error is smaller than that of vanilla TF, indicating that the multi-scale mining mechanism achieves a better distributed expression of the multi-sensor spatiotemporal feature. The MPE of the multilevel fusion TF is greatly reduced compared with the others, indicating that the attention of external factors modifies the degree of prognostic radicalization. By comparing Figure 9a with Figure 9b, it can be found that TF cannot properly learn the trend of the RUL when QUAV is fully loaded (high in power consumption). To explain the phenomenon, the semi-cycle RUL curve is drawn. The curve value represents the RUL when the current maneuver remains unchanged after the current time. The power consumption at the early stage of flight is higher than at the later stage. If the maneuver is continued, the drone very easily loses the mission execution ability due to the input saturation being reached. Consequently, the semi-cycle RUL will be lower than the full-cycle RUL to some extent. The above results in a relatively large prediction error of the network at the early stage of flight. However, different from TF, the sequence prediction results of the proposed method basically follow the objective truth that the life decreases with time under small maneuver changes. In the meantime, this accounts for its relatively large MSE.

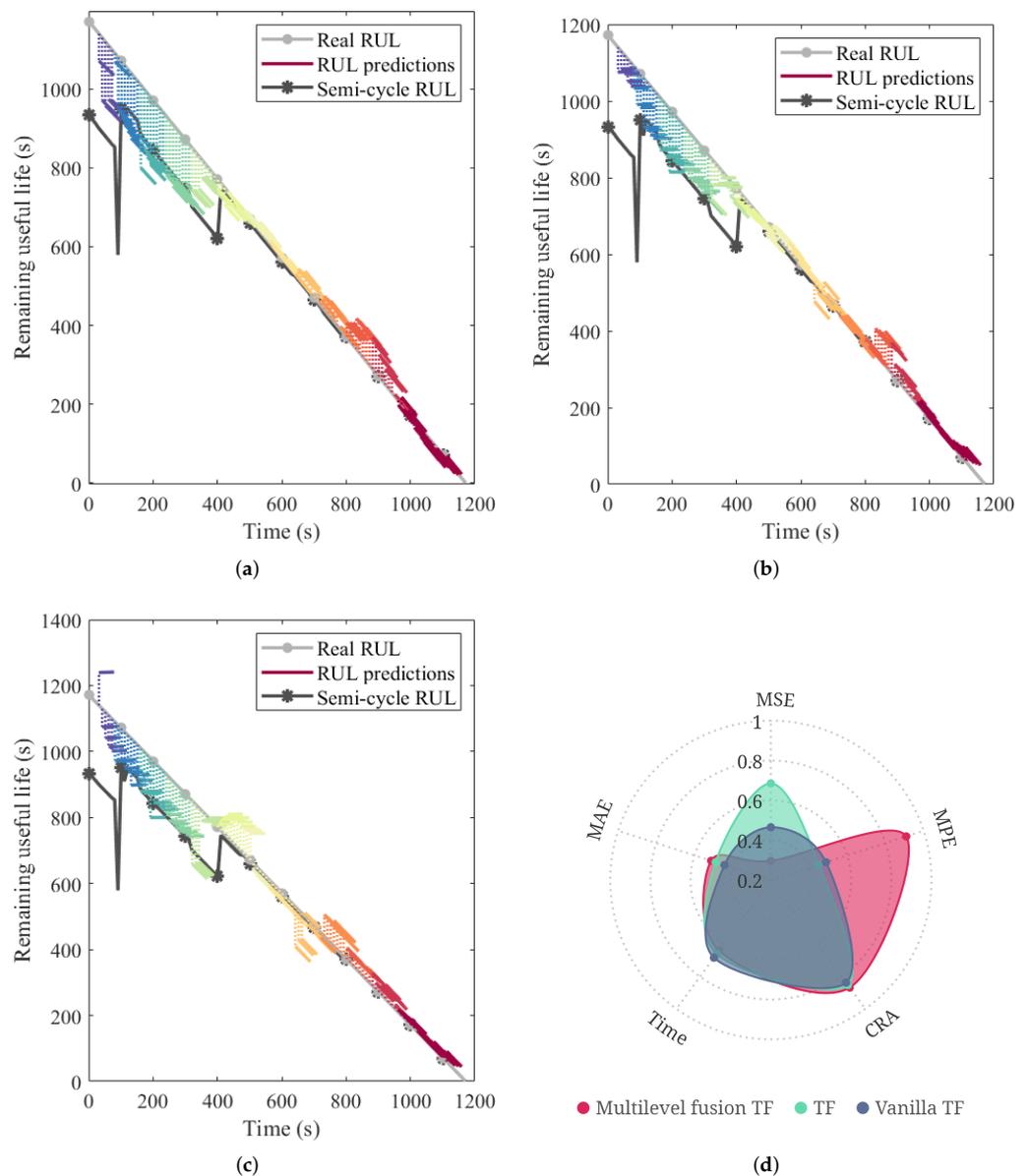


Figure 9. Predictions for the next 48 s. (a) RUL predictions of the multilevel fusion transformer network. (b) RUL predictions of the transformer network. (c) RUL predictions of the vanilla transformer network. (d) The 5-dimensional radar chart for the performance comparison of the aforementioned prediction methods. The performance depicted in the radar chart ranges from 0 to 1 (from low to high). The colored solid line represents the output value of the RUL prediction sequence in the next 48 s. The point corresponding to the colored dotted line refers to the RUL prediction start time. The network needs to obtain data from 32 s before the start time of the current time as an input. As a result, the RUL is predicted every 10 s after the 32 s blank period to meet the maneuver decision-making needs for the full cycle.

Due to the short prediction period, the variation of the RUL prediction sequence is not obvious. Therefore, to further discuss the effect of external factor fusion on the RUL prediction performance with a long prediction period, the prediction experiments for the next 128 s were performed. The prediction results of multilevel fusion TF and its MPE with various start times are demonstrated in Figure 10. Correspondingly, the results of TF are presented in Figure 11.

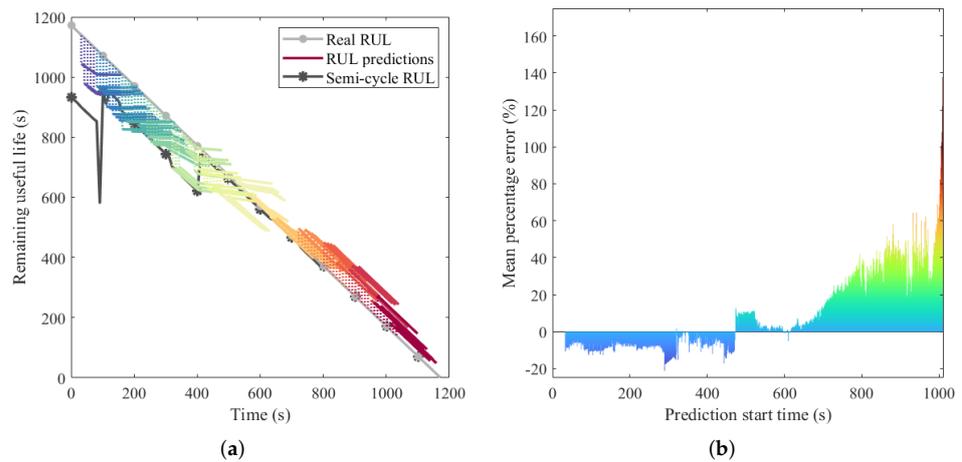


Figure 10. Predictions of the multilevel fusion transformer network for the next 128 s. (a) RUL predictions. (b) The MPE of RUL predictions with different start times. The colored solid line represents the output value of the RUL prediction sequence in the next 128 s. The point corresponding to the colored dotted line refers to the RUL prediction start time. The RUL is predicted every 10 s after the 32 s blank period to meet the maneuver decision-making needs for the full cycle.

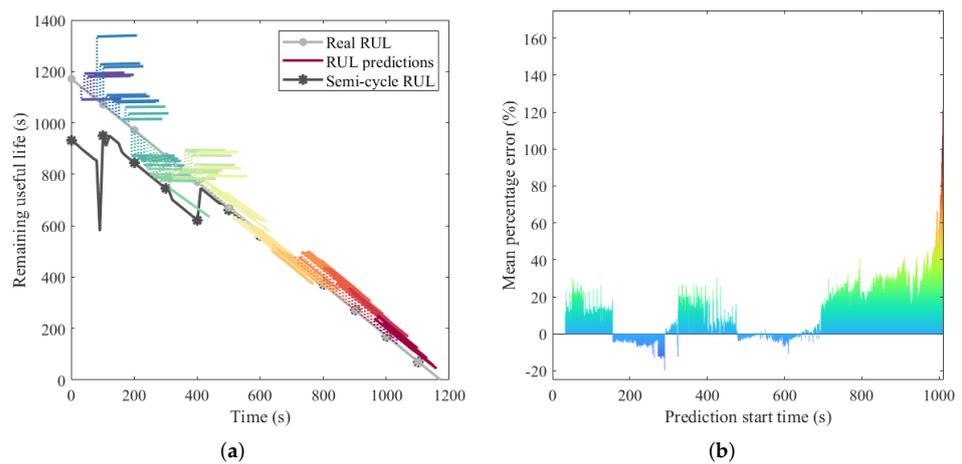


Figure 11. Predictions of the transformer network for the next 128 s. (a) RUL predictions. (b) The MPE of RUL predictions with different start time. The colored solid line represents the output value of the RUL prediction sequence in the next 128s. The point corresponding to the colored dotted line refers to the RUL prediction start time. The RUL is predicted every 10s after the 32s blank period to meet the maneuver decision-making needs for the full cycle.

Figures 10a and 11a show that when the future maneuver changes strongly, the proposed method can change the prediction output after the maneuver change in real-time. It proves that the external factor fusion mechanism can make the TF perceive the performance changes of the highly maneuverable QUAV in the future. Accordingly, either the operator or fault-tolerant controller can adjust the mission planning in time. Moreover, Figures 10b and 11b show that the proposed method has a smaller MPE, and its predicted RUL is more reliable in the highly maneuverable flight.

In addition, according to the performance metrics in Table 8 and the visualization in Figure 12, the multilevel fusion TF can realize the high-precision RUL sequence-to-sequence prediction.

Table 8. Performance of the RUL prediction for the next 128 s.

Metric	Multilevel Fusion TF		TF	
	Single Flight	100 Flights	Single Flight	100 Flights
MSE (s ²)	5.0677×10^3	$(5.0564 \pm 0.2416) \times 10^3$	7.7484×10^3	$(7.4859 \pm 0.3141) \times 10^3$
MPE (%)	8.7461	7.6482 ± 1.1790	14.9258	13.8908 ± 1.4337
CRA	0.8335	0.8428 ± 0.0064	0.8249	0.8315 ± 0.0068
MAE (s)	60.5949	59.7938 ± 0.7437	65.6041	64.3956 ± 1.5350
Time (ms)	80.385		72.907	

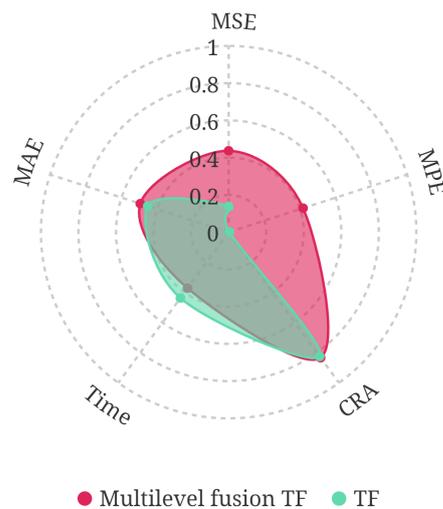


Figure 12. The 5-dimensional radar chart for the performance comparison of predictions for the next 128 s. The performance depicted in the radar chart ranges from 0 to 1 (from low to high).

The disturbance and noise in the highly maneuverable flight, along with the randomness of the deep learning algorithm, expose uncertainties [34]. Differences in a small range can be found in the real RUL and predictions among each flight with the same scheduled flight plan. Therefore, the simulation with the same plan, as shown in the figure, was run 100 times, and the prediction results are also recorded in Table 8. To facilitate observation and comparison, the statistical distribution of prediction performance after normalization is plotted in Figure 13. It can be seen that the prediction performance of the multilevel fusion TF is generally better than TF, and the fluctuation is smaller, which proves its stronger robustness.

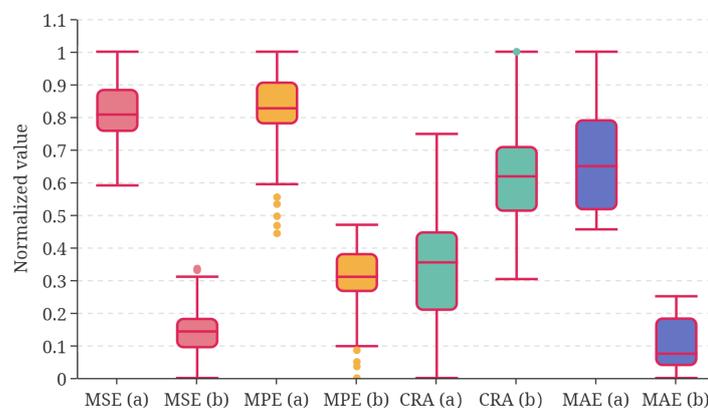


Figure 13. Result of RUL prediction for the next 128 s with disturbance and noise. *a* and *b* represent the TF and multilevel fusion TF, respectively.

6. Conclusions and Future Works

In this paper, a highly maneuverable unmanned aerial vehicle RUL sequence-to-sequence prediction algorithm based on the multilevel fusion transformer network was proposed. This method is an end-to-end deep learning method, which reduces the subjectivity and professional knowledge requirements of feature extraction in the machine learning method. By considering the predictable external factors, this method can realize a sequence-to-sequence RUL prediction and predict future variations of the RUL. It has the guiding significance for measuring the residual flight capability and mission capability of the highly maneuverable UAV. The QUAV model based on actual flight fitted parameters and the lithium-polymer battery discharge mechanism was used for simulation verification, and the results show that the following novel contributions were accomplished:

1. The sequence-to-sequence RUL prediction was realized based on the transformer network's encoder–decoder structure, and its feasibility and real-time performance were validated.
2. A multi-scale feature mining mechanism was designed to achieve the feature layer fusion of multi-sensor signals and the distributed expression of spatio-temporal features. Its effectiveness in improving the precision of end-to-end prediction was illustrated.
3. The external factor fusion layer was constructed to enhance the attention of the algorithm to the predictable information with the future time scale. For the RUL prognostic demand of the highly maneuverable UAV, the superiority of the multilevel fusion transformer network was proven.

While the influence of external factors was considered in this paper, a limitation still exists: the influence of the temperature change on the battery discharge was ignored. At the same time, after long-term use, the battery's internal resistance gradually increases. As a result, future research will focus on the further optimization of the battery discharge model.

Author Contributions: Conceptualization, S.A.; methodology, S.A.; software, S.A.; validation, S.A.; formal analysis, S.A.; investigation, S.A.; resources, J.S. and G.C.; data curation, S.A.; writing—original draft preparation, S.A.; writing—review and editing, S.A. and J.S.; visualization, S.A.; supervision, J.S.; project administration, G.C.; funding acquisition, J.S. and G.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National High-Tech Research and Development Program of China under Grant No. 111GFTQ2019115006 and the National Natural Science Foundation of China under Grant Nos. 91646108 and 62073020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors thank Jiaqi Geng from Carnegie Mellon University for his research assistance throughout this study. The authors also appreciate the Associate Editor and the Reviewers for their valuable comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bi, H.; Qi, G.; Hu, J.; Faradja, P.; Chen, G. Hidden and transient chaotic attractors in the attitude system of quadrotor unmanned aerial vehicle. *Chaos Solitons Fractals* **2020**, *138*, 109815. [[CrossRef](#)]
2. Labbadi, M.; Cherkaoui, M. Adaptive fractional-order nonsingular fast terminal sliding mode-based robust tracking control of quadrotor UAV with Gaussian random disturbances and uncertainties. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2265–2277. [[CrossRef](#)]
3. Labbadi, M.; Boukal, Y.; Cherkaoui, M. Path following control of quadrotor UAV with continuous fractional-order super twisting sliding mode. *J. Intell. Rob. Syst.* **2020**, *100*, 1429–1451. [[CrossRef](#)]
4. Wang, D.; Kong, J.; Zhao, Y.; Tsui, K.L. Piecewise model based intelligent prognostics for state of health prediction of rechargeable batteries with capacity regeneration phenomena. *Measurement* **2019**, *147*, 106836. [[CrossRef](#)]
5. Sierra, G.; Orchard, M.; Goebel, K.; Kulkarni, C. Battery health management for small-size rotary-wing electric unmanned aerial vehicles: An efficient approach for constrained computing platforms. *Reliab. Eng. Syst. Saf.* **2019**, *182*, 166–178. [[CrossRef](#)]

6. Li, X.; Ma, Y.; Zhu, J. An online dual filters RUL prediction method of lithium-ion battery based on unscented particle filter and least squares support vector machine. *Measurement* **2021**, *184*, 109935. [[CrossRef](#)]
7. Wang, B.; Liu, D.; Wang, W.; Peng, X. A hybrid approach for UAV flight data estimation and prediction based on flight mode recognition. *Microelectron. Reliab.* **2018**, *84*, 253–262. [[CrossRef](#)]
8. Tang, X.; Zou, C.; Yao, K.; Lu, J.; Xia, Y.; Gao, F. Aging trajectory prediction for lithium-ion batteries via model migration and Bayesian Monte Carlo method. *Appl. Energy* **2019**, *254*, 113591. [[CrossRef](#)]
9. Jung, S.; Jeong, H. Extended kalman filter-based state of charge and state of power estimation algorithm for unmanned aerial vehicle Li-Po battery packs. *Energy* **2017**, *10*, 1237. [[CrossRef](#)]
10. Baptista, M.L.; Henriques, E.M.; Prendinger, H. Classification prognostics approaches in aviation. *Measurement* **2021**, *182*, 109756. [[CrossRef](#)]
11. Rezaeianjouybari, B.; Shang, Y. Deep learning for prognostics and health management: State of the art, challenges, and opportunities. *Measurement* **2020**, *163*, 107929. [[CrossRef](#)]
12. Sabanci, K. Artificial intelligence based power consumption estimation of two-phase brushless DC motor according to FEA parametric simulation. *Measurement* **2020**, *155*, 107553. [[CrossRef](#)]
13. Miao, H.; Li, B.; Sun, C.; Liu, J. Joint learning of degradation assessment and RUL prediction for aeroengines via dual-task deep LSTM networks. *IEEE Trans. Ind. Inf.* **2019**, *15*, 5023–5032. [[CrossRef](#)]
14. Sarkar, S.; Totaro, M.W.; Kumar, A. An Intelligent Framework for Prediction of a UAV's Flight Time. In Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina del Rey, CA, USA, 25–27 May 2020; pp. 328–332. [[CrossRef](#)]
15. Al-Dulaimi, A.; Zabihi, S.; Asif, A.; Mohammadi, A. A multimodal and hybrid deep neural network model for remaining useful life estimation. *Comput. Ind.* **2019**, *108*, 186–196. [[CrossRef](#)]
16. Chen, W.; Chen, W.; Liu, H.; Wang, Y.; Bi, C.; Gu, Y. A RUL Prediction Method of Small Sample Equipment Based on DCNN-BiLSTM and Domain Adaptation. *Mathematics* **2022**, *10*, 1022. [[CrossRef](#)]
17. Zhao, R.; Wang, D.; Yan, R.; Mao, K.; Shen, F.; Wang, J. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1539–1548. [[CrossRef](#)]
18. Liang, Y.; Ke, S.; Zhang, J.; Yi, X.; Zheng, Y. Geoman: Multi-level attention networks for geo-sensory time series prediction. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; Volume 2018, pp. 3428–3434. [[CrossRef](#)]
19. Ai, S.; Song, J.; Cai, G. A real-time fault diagnosis method for hypersonic air vehicle with sensor fault based on the auto temporal convolutional network. *Aerosp. Sci. Technol.* **2021**, *119*, 107220. [[CrossRef](#)]
20. Song, Y.; Gao, S.; Li, Y.; Jia, L.; Li, Q.; Pang, F. Distributed attention-based temporal convolutional network for remaining useful life prediction. *IEEE IoT J.* **2020**, *8*, 9594–9602. [[CrossRef](#)]
21. Tetko, I.V.; Karpov, P.; Van Deursen, R.; Godin, G. State-of-the-art augmented NLP transformer models for direct and single-step retrosynthesis. *Nat. Commun.* **2020**, *11*, 5575. [[CrossRef](#)]
22. Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; Feichtenhofer, C. Multiscale vision transformers. *arXiv* **2021**, arXiv:2104.11227.
23. Giuliani, F.; Hasan, I.; Cristani, M.; Galasso, F. Transformer networks for trajectory forecasting. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 10335–10342. [[CrossRef](#)]
24. Mo, Y.; Wu, Q.; Li, X.; Huang, B. Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit. *J. Manuf.* **2021**, *32*, 1997–2006. [[CrossRef](#)]
25. Eleftheroglou, N.; Mansouri, S.S.; Loutas, T.; Karvelis, P.; Georgoulas, G.; Nikolakopoulos, G.; Zarouchas, D. Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the Lithium-Polymer batteries of unmanned aerial vehicles with uncertainty quantification. *Appl. Energy* **2019**, *254*, 113677. [[CrossRef](#)]
26. Cui, G.; Yang, W.; Yu, J.; Li, Z.; Tao, C. Fixed-Time Prescribed Performance Adaptive Trajectory Tracking Control for a QUAV. *IEEE Trans. Circuits. Syst. II Express Briefs* **2021**, *69*, 494–498. [[CrossRef](#)]
27. Mansouri, S.S.; Karvelis, P.; Georgoulas, G.; Nikolakopoulos, G. Remaining useful battery life prediction for UAVs based on machine learning. *IFAC-PapersOnLine* **2017**, *50*, 4727–4732. [[CrossRef](#)]
28. Dai, X.; Quan, Q.; Ren, J.; Cai, K.Y. An analytical design-optimization method for electric propulsion systems of multicopter UAVs with desired hovering endurance. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 228–239. [[CrossRef](#)]
29. Dai, X.; Ke, C.; Quan, Q.; Cai, K. RFLySim: Automatic test platform for UAV autopilot systems with FPGA-based hardware-in-the-loop simulations. *Aerosp. Sci. Technol.* **2021**, *114*, 106727. [[CrossRef](#)]
30. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
31. Ai, S.; Shang, W.; Song, J.; Cai, G. Fault Diagnosis of the Four-Rotor Unmanned Aerial Vehicle using the Optimized Deep Forest Algorithm based on the Wavelet Packet Translation. In Proceedings of the 2021 8th International Conference on Dependable Systems and Their Applications (DSA), Yinchuan, China, 5–6 August 2021; pp. 581–589. [[CrossRef](#)]
32. Eleftheroglou, N.; Zarouchas, D.; Loutas, T.; Alderliesten, R.; Benedictus, R. Structural health monitoring data fusion for in-situ life prognosis of composite structures. *Reliab. Eng. Syst. Saf.* **2018**, *178*, 40–54. [[CrossRef](#)]
33. Kong, J.; Wang, D.; Yan, T.; Zhu, J.; Zhang, X. Accelerated Stress Factors based Nonlinear Wiener Process Model for Lithium-ion Battery Prognostics. *IEEE Trans. Ind. Electron.* **2021**. [[CrossRef](#)]
34. Scardapane, S.; Wang, D. Randomness in neural networks: An overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2017**, *7*, e1200. [[CrossRef](#)]