This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

Digital Twin-Guided Spatiotemporal Graph Representation Learning for Reliable UAV Fault Diagnosis under Complex Flight Conditions

Jinhu Tu, Xiaohong Nian, and Xunhua Dai

***Abstract*—In low-altitude smart city operations and high-maneuverability missions, unmanned aerial vehicle (UAV) fault diagnosis is challenged by strong dynamic disturbances, tightly coupled sensors, and frequent shifts in operating conditions, often causing feature confusion and reduced generalization in existing data-driven methods. To address this, we propose the Twin-Consistent Spatiotemporal Graph Diagnosis Network (TC-SGDN), a digital twin (DT)-driven framework centered on dynamic health consistency. Virtual twin nodes alongside physical sensors form a semantically enriched spatiotemporal graph capturing temporal dependencies, sensor interactions, and corrective relationships, enabling joint modeling of system dynamics and faults. A twin-consistency-based attention and temporal encoding mechanism further suppresses non-fault disturbances from aggressive maneuvers. Experiments on real and simulated UAV datasets covering diverse conditions and fault types show that TC-SGDN significantly improves accuracy, robustness, and cross-condition generalization, demonstrating that embedding digital twins as dynamic health baselines enhances reliability and adaptability, offering a generalizable paradigm for intelligent UAV health management. The benchmark data and implementation code are publicly available at https://github.com/tujinhu/DT-DiagnosisV/tree/main.**

***Index Terms*—Unmanned aerial vehicles (UAVs), intelligent fault diagnosis, graph attention network (GAT), digital twin (DT), spatiotemporal features**

I. Introduction

A. Background

With the opening of low-altitude airspace to smart cities, emergency response, logistics, and infrastructure inspection, unmanned aerial vehicles (UAVs) are increasingly deployed in complex and safety-critical missions [1]. Ensuring reliable operation under dynamic and uncertain environments has therefore become a fundamental challenge, with intelligent fault diagnosis recognized as a key enabling technology for autonomous UAV safety [2], [3], [4], [5].

Unlike conventional industrial systems, UAVs exhibit strong nonlinear dynamics, tight sensor-actuator coupling, and pronounced condition-dependent behaviors. Flight states are jointly affected by maneuver intensity, aerodynamic disturbances, actuator interactions, and environmental uncertainties, making reliable fault diagnosis particularly challenging. During aggressive maneuvers such as sharp turns, rapid acceleration, emergency braking, and wind disturbances, transient fluctuations may resemble genuine fault signatures. The central challenge is therefore to distinguish fault-induced anomalies from maneuver-driven normal dynamics under varying operating conditions.

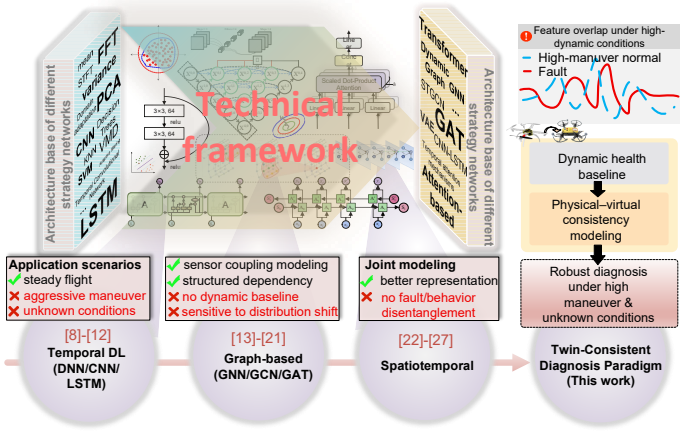


Fig. 1. Mainstream paradigms and limitations of UAV fault diagnosis. Three core paradigms (temporal-based, graph-based modeling, and spatiotemporal learning) are presented. Existing spatiotemporal learning methods still face a critical issue in high-dynamic scenarios: the lack of an adaptive health baseline leads to feature overlap between normal maneuvers and faults, which drives our digital twin-enabled consistent diagnosis paradigm.

B. Related work and motivation

With growing UAV mission complexity and richer sensor data, fault diagnosis has shifted from traditional signal- and physics-based methods to data-driven deep learning approaches [6]. Early studies employed deep neural networks (DNNs), convolutional neural networks (CNNs), and recurrent networks to extract patterns from time-series sensor data [7], [8], [9], [10], [11]. More recent intelligent fault diagnosis studies have further improved feature representation, interpretability, and generalization. For example, deep adversarial capsule networks exploit capsule representation and adversarial learning for compound fault decoupling and multidomain generalization [12], while WavCapsNet combines wavelet-kernel feature extraction with capsule routing and backward tracking to enhance interpretable compound fault diagnosis

This work was supported by the National Natural Science Foundation of China under Grant 62406345 and the Hunan Provincial Natural Science Foundation of China (2025JJ50341).

Jinhu Tu and Xiaohong Nian are with the Department of School of Automation, Central South University, Changsha, 410083, China (Email: tjhcsu@csu.edu.cn, xhnian@csu.edu.cn)

Xunhua Dai (Corresponding author) is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (Email: dai.xh@csu.edu.cn).

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

[13]. Transformer-based methods, such as the Exofeature-Aware Transformer, further model state-feature dependencies through tailored embedding and attention mechanisms [14]. These studies show that intelligent diagnosis is moving beyond simple temporal feature extraction toward structured, robust, and interpretable representation learning. However, they still primarily infer health states from measured signal patterns or learned latent features, without constructing an online condition-adaptive physical reference for the expected system response.

To better capture structural relationships within complex systems, graph neural networks (GNNs) have recently attracted increasing attention in fault diagnosis [15]. In these approaches, sensors or feature components are represented as graph nodes and their interactions as edges, enabling explicit modeling of inter-sensor or inter-feature dependencies [16], [17], [18]. Building on this idea, recent studies have introduced dynamic graph structures, multiscale graph representations, attention-based graph learning mechanisms, and time-frequency graph models to improve adaptability under varying operating conditions [19], [20], [21], [22], [23], [24]. Nevertheless, most existing graph structures are still constructed from statistical correlations, learned feature dependencies, or predefined topologies. They rarely incorporate a synchronized healthy-reference trajectory as a graph-native constraint, and therefore cannot explicitly distinguish expected maneuver-induced dynamics from fault-induced physical-virtual deviations.

Meanwhile, as UAV operating environments become increasingly dynamic, joint spatial-temporal modeling has consistently outperformed purely spatial or temporal architectures [25], [26]. Recent studies have introduced spatiotemporal graph encoders, hypergraph representations, and hybrid graph-temporal frameworks integrating GNNs with recurrent or convolutional networks [27], [28], [29], [30]. Despite improved representation capacity, these methods still mainly learn mappings between observed sensor patterns and predefined fault labels. Their performance therefore remains dependent on training distributions and may degrade under unseen maneuvers, strong disturbances, and out-of-distribution flight conditions. The key limitation is not simply insufficient model complexity, but the absence of a dynamic health baseline describing what the UAV should do under the same control inputs and flight conditions. Without such a reference, highly dynamic yet normal responses can overlap with true fault signatures, reducing robustness and generalization.

Digital twin (DT) technology offers a promising solution to this challenge [31]. By maintaining a synchronized virtual counterpart of the physical UAV, DTs can generate expected system responses under varying conditions and continuously compare physical and simulated behaviors [32], [33]. Prior studies have demonstrated DT-assisted diagnosis for improving robustness and data efficiency [34], [35]. Recent advances further extend DT-driven diagnosis through DT-assisted data fusion and domain adaptation for imbalanced fault recognition [36], physics-consistent fault synthesis via DT-constrained generative models [37], IMU fault diagnosis using DT residuals [38], and PX4-based DT platforms for UAV fault simulation and validation [39]. Nevertheless, most existing frameworks still employ DTs primarily as external auxiliary modules for offline simulation, synthetic data generation, parameter estimation, or post-hoc validation, rather than integrating them into the diagnostic representation learning process itself. Consequently, DTs rarely participate directly in graph construction or spatiotemporal representation learning, and most existing approaches still diagnose faults by fitting correlations between observed sensor patterns and fault labels. As a result, DT consistency is seldom treated as a graph-native structural constraint for modeling deviations from continuously evolving healthy UAV behaviors.

Fig. 1 illustrates the evolution of UAV fault diagnosis from temporal modeling to graph-based spatiotemporal learning. Despite increasing representational power, most approaches remain observation-driven, inferring faults solely from measured signals without modeling the healthy response under identical maneuvers. To address this gap, we embed synchronized DT states into graph-native spatiotemporal learning as online health references rather than external simulation tools, reformulating diagnosis as physical–virtual consistency deviation modeling. By continuously comparing physical observations with synchronized healthy twin responses, the proposed framework establishes an adaptive health baseline that disentangles maneuver-induced dynamics from genuine fault anomalies under complex and unseen flight conditions.

C. Research problems and contributions

Based on the above motivation, this paper investigates the following research problems:

i) How to construct synchronized DT-driven dynamic health baselines for reliable fault discrimination under highly varying flight conditions?
ii) How to embed DT consistency directly into graph-native spatiotemporal learning, enabling consistency-aware graph construction and semantic representation?
iii) How to develop a physically grounded diagnosis framework that remains robust and generalizable under aggressive maneuvers, disturbances, synchronization uncertainty, and out-of-distribution (OOD) conditions?

To address these challenges, this paper proposes the Twin-Consistent Spatiotemporal Graph Diagnosis Network (TC-SGDN), a DT-native graph framework for UAV fault diagnosis. TC-SGDN embeds synchronized DT responses into graph representation learning as online health references and constructs physics-semantic relations among temporal dynamics, sensor coupling, and physical-virtual residuals. This design enables consistency-aware spatiotemporal diagnosis under dynamic and previously unseen flight conditions. The main contributions of this paper are:

i) A DT-native UAV fault diagnosis paradigm that embeds synchronized DT consistency into spatiotemporal graph learning, reformulating diagnosis as physical-virtual consistency deviation modeling.
ii) A physics-semantic spatiotemporal graph jointly encoding temporal dependencies, sensor coupling, and DT-

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

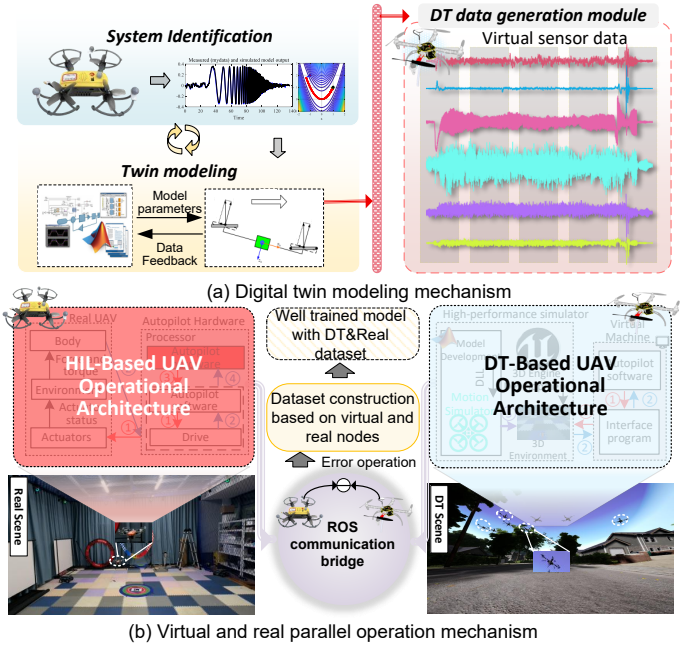


Fig. 2. Architecture of the real-virtual UAV node pair and the DT modeling framework. The DT, whose component-level mechanism modeling was developed in our previous work [40] and released as an open-source platform at https://github.com/RflySim/RflySimDT, is treated here as a high-fidelity virtual counterpart running in parallel with the physical UAV. In this study, the DT is integrated into the fault diagnosis pipeline to enable real-time physical-virtual response alignment and provide a dynamic health reference for subsequent spatiotemporal graph reasoning.

consistency deviations under highly dynamic flight conditions.

iii) A DT-consistent diagnosis framework that improves robustness and OOD generalization under aggressive maneuvers and disturbances via condition-adaptive health references rather than increased model complexity.

iv) A comprehensive open-source UAV fault diagnosis benchmark covering diverse flight conditions, maneuvers, sensor configurations, and fault types for reproducible evaluation.

D. Organization

The remainder of this paper is organized as follows. Section II reviews the multi-head graph attention network (GAT). Section III presents the proposed TC-SGDN framework, including DT-based virtual-physical synchronization, DT-consistent graph embedding, rule-guided attention, and spatiotemporal consistency learning. Section IV reports Vir2Vir, Rea2Vir, and real-world quadcopter flight validations. Finally, Section V concludes the paper and outlines future work.

II. Preliminaries

This section introduces the GAT formulation used as the basic graph learning operator in the proposed framework. A graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{H})$, where $\mathcal{V} = \{v_i\}_{i=1}^{n}$ is the node set, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the edge set, $\mathbf{A} = [A_{ij}] \in \mathbb{R}^{n\times n}$ is the adjacency or edge-weight matrix, and $\mathbf{H} = [\mathbf{h}_1, \cdots, \mathbf{h}_n]^{\mathrm{T}} \in \mathbb{R}^{n\times d_{\mathrm{in}}}$ is the node feature matrix. Here, n denotes the number of nodes, d_{in} is the input feature dimension, and $\mathbf{h}_i \in \mathbb{R}^{d_{\mathrm{in}}}$ is the feature vector of node v_i. The objective of multi-head GAT is to map $\mathbf{H}$ to an enhanced representation $\mathbf{H}'$. For head $l \in \{1, \cdots, L\}$, the linear projection of node v_i is

$$\mathbf{h}_i'^{(l)} = \mathbf{W}^{(l)}\mathbf{h}_i, \quad \mathbf{W}^{(l)} \in \mathbb{R}^{d_{\mathrm{out}}\times d_{\mathrm{in}}}, \quad \mathbf{h}_i'^{(l)} \in \mathbb{R}^{d_{\mathrm{out}}}, \tag{1}$$

where L is the number of attention heads and d_{out} is the output feature dimension of each head. For an edge $\langle v_i, v_j\rangle \in \mathcal{E}$, the unnormalized attention logit of head l is computed as

$$s_{ij}^{(l)} = \mathbf{a}^{(l)^{\mathrm{T}}}\left[\mathbf{h}_i'^{(l)} \| \mathbf{h}_j'^{(l)}\right], \quad \mathbf{a}^{(l)} \in \mathbb{R}^{2d_{\mathrm{out}}}, \tag{2}$$

where $\|$ denotes feature concatenation and $\mathbf{a}^{(l)}$ is a trainable attention vector. The normalized attention coefficient is

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\mathrm{LeakyReLU}(s_{ij}^{(l)})\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}(s_{ik}^{(l)})\right)}, \tag{3}$$

where $\mathcal{N}_i = \{v_k \mid \langle v_i, v_k\rangle \in \mathcal{E}\}$ is the neighbor set of v_i. The head-specific aggregated feature is

$$\mathbf{\Gamma}_i^{(l)} = \sigma\left(\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(l)}\mathbf{h}_j'^{(l)}\right), \tag{4}$$

where $\sigma(\cdot)$ is the activation function. The final multi-head representation is

$$\mathbf{h}_i'' = \begin{cases} \frac{1}{L}\sum_{l=1}^{L}\mathbf{\Gamma}_i^{(l)}, & \text{mean aggregation,} \\ \mathbf{\Gamma}_i^{(1)}\|\cdots\|\mathbf{\Gamma}_i^{(L)}, & \text{concatenation aggregation.} \end{cases} \tag{5}$$

III. Methodology

A. Overview of TC-SGDN framework

This section presents TC-SGDN, which integrates synchronized DT references, physics-semantic graph construction, rule-guided attention, and temporal diagnosis learning into a unified framework. The framework consists of four components:

i) a synchronized DT mechanism providing condition-adaptive healthy references;

ii) a DT-consistent graph embedding strategy integrating temporal dynamics, sensor coupling, and physical-virtual consistency relationships;

iii) a physics-semantic rule-based attention mechanism for consistency-aware graph propagation, and

iv) a spatiotemporal consistency learning network for unified spatial-temporal diagnosis modeling.

B. DT mechanism

TC-SGDN uses a synchronized high-fidelity DT to provide healthy reference responses for physical UAV measurements. These responses are aligned with real sensor streams and later converted into physical-virtual residuals for graph construction.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

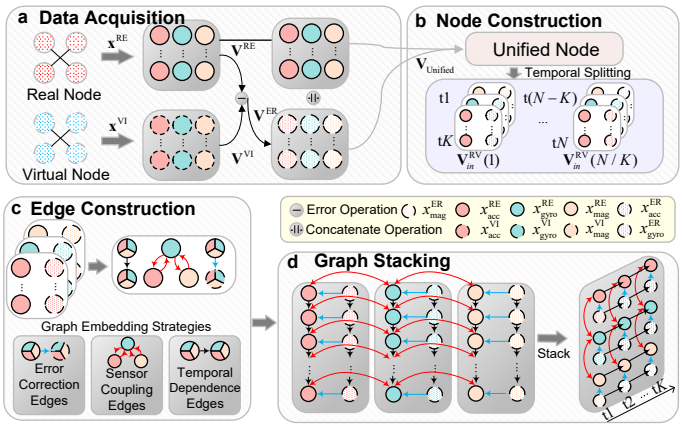


Fig. 3. DT-consistent graph embedding workflow. (a) Physical and virtual measurements are aligned to construct residual features. (b) Real and residual nodes are organized within each temporal window. (c) Temporal, sensor-coupling, and error-correction edges encode physics-semantic relations. (d) The resulting graph sequence is stacked as the input to TC-SGDN.

TABLE I
SYMBOL ANNOTATION FOR UAV GNN APPLICATION BASED ON DT

Symbols	Description
$\mathcal{G}$	DT-consistent spatiotemporal graph used as the diagnostic input.
$\mathcal{V}$	Graph node set, including real sensor nodes and DT residual nodes. This process is illustrated in Fig.3(a) and Fig.3(b) and detailed in Section III-D.
$\mathcal{E}$	Graph edge set, including temporal dependency edges, sensor coupling edges, and DT-consistency error correction edges, as illustrated in Fig.3(c) and detailed in Section III-E.
$\mathbf{A}$	Adjacency or edge-weight matrix derived from $\mathcal{E}$.
$\mathbf{H}$	Node feature matrix used by the graph neural network.
$\mathbf{v}_s^{\mathrm{RE}}(t)$	Real sensor node feature of sensor s at time step t.
$\mathbf{v}_s^{\mathrm{ER}}(t)$	DT residual node feature, defined as the physical-virtual deviation of sensor s at time step t.

1) Twin modeling mechanism: As shown in Fig.2(a), virtual nodes replicate physical UAV states via a high-fidelity DT grounded in system identification, as implemented in our work [40]. Unlike a simple simulator, the twin maps dynamics, sensors, and actuators to provide a pristine "expected response" baseline. At its core, a 100 Hz six-degree-of-freedom rigid-body model captures UAV motion under control inputs:

$$\begin{aligned}
{}^{\mathrm{e}}\dot{\mathbf{p}} &= {}^{\mathrm{e}}\mathbf{v}, \\
{}^{\mathrm{e}}\mathbf{v} &= \mathbf{R}_{\mathrm{be}}^{\mathrm{T}}\,{}^{\mathrm{b}}\mathbf{v}, \\
{}^{\mathrm{b}}\dot{\mathbf{v}} &= -[{}^{\mathrm{b}}\boldsymbol{\omega}]_{\times}\,{}^{\mathrm{b}}\mathbf{v} + g\mathbf{R}_{\mathrm{be}}^{\mathrm{T}}\mathbf{e}_3 + \frac{\mathbf{F}}{m_{\mathrm{uav}}}, \\
\dot{\mathbf{q}} &= \frac{1}{2}\begin{bmatrix} 0 & -{}^{\mathrm{b}}\boldsymbol{\omega}^{\mathrm{T}} \\ {}^{\mathrm{b}}\boldsymbol{\omega} & -[{}^{\mathrm{b}}\boldsymbol{\omega}]_{\times} \end{bmatrix}\mathbf{q}, \\
\mathbf{J}\,{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} &= -{}^{\mathrm{b}}\boldsymbol{\omega} \times (\mathbf{J}\,{}^{\mathrm{b}}\boldsymbol{\omega}) + \boldsymbol{\tau}_{\mathrm{uav}} + \mathbf{G}_a,
\end{aligned} \tag{6}$$

where ${}^{\mathrm{e}}\mathbf{p}, {}^{\mathrm{e}}\mathbf{v} \in \mathbb{R}^3$ denote the UAV position and velocity in the world frame, and ${}^{\mathrm{b}}\mathbf{v}$ denotes the velocity expressed in the body frame. The matrix $\mathbf{R}_{\mathrm{be}}^{\mathrm{T}}$ is the rotation matrix. The notation $[\cdot]_{\times}$ denotes the skew-symmetric matrix for the cross product. The vector $\mathbf{q}$ is the attitude quaternion, $\mathbf{e}_3 = [0,0,1]^{\mathrm{T}}$, m_{uav} is the UAV mass, $\mathbf{F}$ is the total rotor thrust vector, ${}^{\mathrm{b}}\boldsymbol{\omega}$ is the body angular velocity, g is the gravitational acceleration, $\boldsymbol{\tau}_{\mathrm{uav}} \in \mathbb{R}^3$ is the rotor-generated torque vector, $\mathbf{J} \in \mathbb{R}^{3\times3}$ is the inertia matrix, and $\mathbf{G}_a \in \mathbb{R}^3$ is the gyroscopic torque.

The DT provides a dynamically consistent healthy reference rather than an auxiliary data source. Under shared control input $\mathbf{u}(t)$, the physical response is decomposed as $\mathbf{x}_s^{\mathrm{RE}}(t) = \mathbf{x}_s^0(t;\mathbf{u}) + \mathbf{d}_s(t) + \boldsymbol{\eta}_s(t)$, where $\mathbf{x}_s^0(t;\mathbf{u})$ is the nominal healthy response, $\mathbf{d}_s(t)$ is the fault-induced deviation, and $\boldsymbol{\eta}_s(t)$ denotes measurement noise and disturbances. Since the synchronized DT output $\mathbf{x}_s^{\mathrm{VI}}(t)$ approximates $\mathbf{x}_s^0(t;\mathbf{u})$, the residual $\mathbf{x}_s^{\mathrm{RE}}(t) - \mathbf{x}_s^{\mathrm{VI}}(t)$ suppresses maneuver-induced dynamics while preserving fault-related deviations. This residual-consistency view motivates embedding DT virtual responses as condition-adaptive references in graph reasoning.

2) Virtual-physical synchronization: As shown in Fig. 2(b), the physical UAV and DT operate in parallel under shared control inputs, while real sensor measurements and DT responses are paired through timestamp alignment. The goal of synchronization is to construct bounded physical-virtual pairs from which reliable residuals can be computed. We define this mechanism as follows.

Definition 1 (Virtual-physical synchronization). Let $\mathbf{u}(t)$ denote the shared control input applied to both the physical UAV and the DT. For a physical timestamp t_k^{RE}, the nearest DT timestamp is

$$l^*(k) = \arg\min_l \left|t_k^{\mathrm{RE}} - t_l^{\mathrm{VI}}\right|, \quad \Delta t_k = \left|t_k^{\mathrm{RE}} - t_{l^*(k)}^{\mathrm{VI}}\right|. \tag{7}$$

A physical-virtual pair is regarded as synchronized only if the alignment error satisfies $\Delta t_k \le \Delta_{\mathrm{sync}}$, where Δ_{sync} is the synchronization tolerance determined by the sensing rate and communication latency of the DT platform. For each admissible timestamp, the aligned DT response is obtained by nearest-neighbor matching or local interpolation, $\bar{\mathbf{x}}_s^{\mathrm{VI}}(t_k^{\mathrm{RE}}) = \mathcal{I}_s\left(t_k^{\mathrm{RE}}\right)$, where $\mathcal{I}_s(\cdot)$ denotes the alignment operator constructed from valid DT samples. The synchronized sample is then defined as

$$\mathcal{P}_k = \left\{\mathbf{x}_s^{\mathrm{RE}}(t_k^{\mathrm{RE}}), \bar{\mathbf{x}}_s^{\mathrm{VI}}(t_k^{\mathrm{RE}}), \mathbf{r}_s^{\mathrm{PV}}(t_k^{\mathrm{RE}})\right\}_{s\in\mathcal{S}}, \tag{8}$$

where $\mathbf{r}_s^{\mathrm{PV}}(t_k^{\mathrm{RE}}) = \mathbf{x}_s^{\mathrm{RE}}(t_k^{\mathrm{RE}}) - \bar{\mathbf{x}}_s^{\mathrm{VI}}(t_k^{\mathrm{RE}})$ is the physical-virtual consistency residual. Therefore, the proposed synchronization mechanism does not require perfectly identical timestamps, but constructs residuals only from bounded-alignment pairs. The complete synchronization and residual-construction procedure is summarized in Algorithm 1.

Remark 1: Unlike static thresholds or offline residual baselines, the synchronized DT reference co-evolves with control inputs and flight dynamics. Consequently, the resulting residuals capture physical-virtual consistency deviations rather than temporal misalignment artifacts. These residual-consistency features are embedded into the spatiotemporal graph as DT-consistency constraints, enabling adaptive discrimination between maneuver-induced normal dynamics and genuine fault anomalies.

C. DT-consistent UAV graph embedding

Building upon Sections II and III-B, TC-SGDN constructs a DT-consistent spatiotemporal graph that unifies temporal dynamics, sensor coupling, and physical–virtual consistency for UAV fault diagnosis. As summarized in Table I, nodes $\mathbf{V}$ combine sensor observations and DT-derived residuals as

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

Algorithm 1 Virtual-Physical Synchronization and Residual Construction

Require: Physical measurements $\{(\mathbf{x}_s^{\mathrm{RE}}(t_k^{\mathrm{RE}}), t_k^{\mathrm{RE}})\}$, DT responses $\{(\mathbf{x}_s^{\mathrm{VI}}(t_l^{\mathrm{VI}}), t_l^{\mathrm{VI}})\}$, shared control command $\mathbf{u}(t)$, tolerance Δ_{sync}
Ensure: Synchronized pairs $\{\mathcal{P}_k\}$ and residual $\{\mathbf{r}_s^{\mathrm{PV}}(k)\}$
for each physical timestamp t_k^{RE} **do**
  Execute the same control command $\mathbf{u}(t_k^{\mathrm{RE}})$ on both the physical UAV and DT
  Find nearest DT timestamp: $l^*(k) = \arg\min_l |t_k^{\mathrm{RE}} - t_l^{\mathrm{VI}}|$
  Compute synchronization error: $\Delta t_k = |t_k^{\mathrm{RE}} - t_{l^*(k)}^{\mathrm{VI}}|$
  if $\Delta t_k \le \Delta_{\mathrm{sync}}$ **then**
    for each sensor $s \in \mathcal{S}$ **do**
      Obtain aligned DT response: $\bar{\mathbf{x}}_s^{\mathrm{VI}}(k) = \mathcal{I}_s\left(t_k^{\mathrm{RE}}\right)$
      Compute physical-virtual residual: $\mathbf{r}_s^{\mathrm{PV}}(k) = \mathbf{x}_s^{\mathrm{RE}}(t_k^{\mathrm{RE}}) - \bar{\mathbf{x}}_s^{\mathrm{VI}}(k)$
    end for
    Construct synchronized sample pair: $\mathcal{P}_k = \left\{\mathbf{x}_s^{\mathrm{RE}}, \bar{\mathbf{x}}_s^{\mathrm{VI}}, \mathbf{r}_s^{\mathrm{PV}}\right\}_{s\in\mathcal{S}}$
  else
    Discard or buffer unsynchronized samples
  end if
end for
return $\{\mathcal{P}_k\}$

health-aware representations, whereas edges $\mathbf{E}$ encode temporal dependencies, sensor interactions, and DT-consistency corrections. These components form $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A}, \mathbf{H})$ for consistency-aware spatiotemporal learning. The graph embedding process in Fig. 3 comprises four stages:

i) Align real UAV observations with synchronized DT outputs and compute physical-virtual residual features.
ii) Construct DT-consistent graph nodes using real sensor states and residual representations.
iii) Define physics-semantic edges based on temporal continuity, sensor coupling relationships, and DT-consistency interactions.
iv) Aggregate node and edge representations to generate the final graph input for consistency-aware spatiotemporal learning.

Specifically, the data originate from the three-axis accelerometer, gyroscope, and magnetometer of the Inertial Measurement Unit (IMU). The lightweight data provided by these sensors effectively reduces computational complexity, ensuring that the system performs well in applications with high real-time requirements [29]. Let $\mathcal{S} = \{\mathrm{acc}, \mathrm{gyro}, \mathrm{mag}\}$ denote the sensor-type set. For sensor $s \in \mathcal{S}$ and synchronized sample index $i \in \{1, \cdots, N\}$, $\mathbf{x}_s^{\mathrm{RE}}(i) \in \mathbb{R}^3$ denotes the real three-axis measurement, and $\mathbf{x}_s^{\mathrm{VI}}(i) \in \mathbb{R}^3$ denotes the timestamp-aligned virtual measurement obtained through Algorithm 1, where N is the number of valid synchronized physical-virtual sample pairs. The full real and virtual datasets are denoted by $\mathbf{X}^{\mathrm{RE}} = \{\mathbf{x}_s^{\mathrm{RE}}(i) \mid s \in \mathcal{S}, i = 1, \cdots, N\}$ and $\mathbf{X}^{\mathrm{VI}} = \{\mathbf{x}_s^{\mathrm{VI}}(i) \mid s \in \mathcal{S}, i = 1, \cdots, N\}$, respectively.

Building upon the DT-consistent graph, UAV fault diagnosis is reformulated as consistency-aware spatiotemporal representation learning. Nodes $\mathbf{V}$ integrate sensor measurements and DT-derived residual semantics, while edges $\mathbf{E}$ encode temporal dependencies, sensor coupling, and physical–virtual consistency. The diagnosis objective is defined as

$$\hat{y} = f_\theta(\mathbf{G}), \quad \hat{y} \in \{1, \ldots, C\}, \tag{9}$$

where $f_\theta(\cdot)$ denotes TC-SGDN and C is the number of health conditions. Experiments consider IMU-based flight data with $C = 4$, including one normal state and three fault categories (Table II). Unlike conventional signal-level classification, inference is performed on DT-consistent graph representations, where residual features provide adaptive healthy references that suppress maneuver-induced variations and emphasize fault-sensitive deviations. Consequently, faults are identified through violations of temporal, structural, or physical–virtual consistency rather than raw signal patterns.

Remark 2: Both real and virtual nodes share the same control input sequence, while their sensor responses are paired via the bounded timestamp alignment rule in *Definition 1* and Algorithm 1. Accordingly, the resulting graph is weakly synchronized: a bounded timing mismatch $\Delta t_k \le \Delta_{\mathrm{sync}}$ is tolerated, while consistency is preserved through the shared control history. This design enables practical deployment under communication latency and sampling mismatch, while maintaining consistency-aware fault discrimination.

D. Graph node construction

To preserve physical-virtual consistency during graph learning, DT-derived residual states are embedded as graph-native nodes together with physical sensor states. For each sensor $s \in \mathcal{S}$ and time sample i, the real node feature and residual node feature are defined as

$$\mathbf{v}_s^{\mathrm{RE}}(i) = \mathbf{x}_s^{\mathrm{RE}}(i), \quad \mathbf{v}_s^{\mathrm{ER}}(i) = \mathbf{x}_s^{\mathrm{RE}}(i) - \mathbf{x}_s^{\mathrm{VI}}(i), \tag{10}$$

where $\mathbf{v}_s^{\mathrm{RE}}(i), \mathbf{v}_s^{\mathrm{ER}}(i) \in \mathbb{R}^3$ and the superscript ER denotes the DT residual/error node. The DT-consistent node set is therefore

$$\mathcal{V}^{\mathrm{RV}} = \{\mathbf{v}_s^r(i) \mid s \in \mathcal{S},\ r \in \{\mathrm{RE}, \mathrm{ER}\},\ i = 1, \cdots, N\}. \tag{11}$$

For the n-th input window with length K, the node subset is $\mathcal{V}_{\mathrm{in}}^{\mathrm{RV}}(n) = \{\mathbf{v}_s^r(t) \mid s \in \mathcal{S},\ r \in \{\mathrm{RE}, \mathrm{ER}\}\}$, where $t = (n-1)K + 1, \cdots, nK$, and K is the number of time steps in each input window.

E. Physics-semantic edge construction

To jointly preserve UAV maneuver continuity, multi-sensor physical interactions, and DT-consistency relationships within a unified graph structure, TC-SGDN defines three complementary physics-semantic edge types within each input window: temporal dependency edges, sensor coupling edges, and DT-consistency error correction edges.

- *Temporal Dependency Edges*: These edges connect the same node type across adjacent time steps and capture the temporal evolution of UAV dynamic states:

$$\mathcal{E}_{\mathrm{T}} = \{\langle \mathbf{v}_s^r(t), \mathbf{v}_s^r(t+1) \rangle \mid s \in \mathcal{S},\ r \in \{\mathrm{RE}, \mathrm{ER}\}\}_{t=1}^{K-1}. \tag{12}$$

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

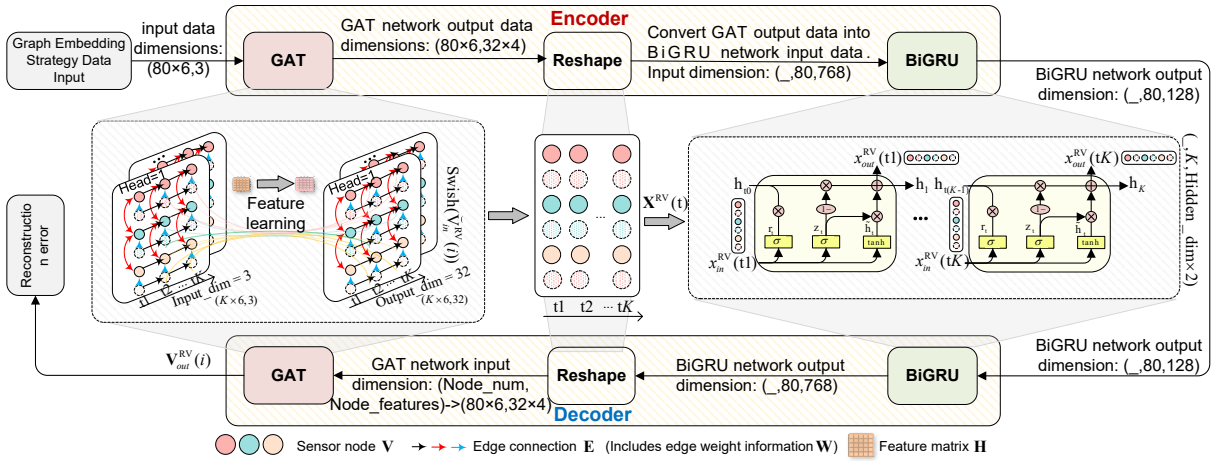


Fig. 4. TC-SGDN architecture.

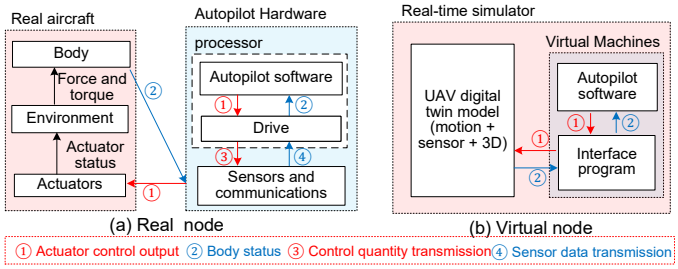


Fig. 5. The underlying architecture diagram of real UAV nodes and virtual UAV nodes

- *Sensor Coupling Edges*: These edges connect different real sensor nodes at the same time step to model physically coupled multi-sensor interactions induced by UAV dynamics:

$$\mathcal{E}_{\mathrm{C}} = \left\{ \left\langle \mathbf{v}_s^{\mathrm{RE}}(t), \mathbf{v}_{s'}^{\mathrm{RE}}(t) \right\rangle \mid s, s' \in \mathcal{S},\ s \neq s' \right\}_{t=1}^{K}. \quad (13)$$

- *Error Correction Edges*: These edges connect each real sensor node with its corresponding DT residual node at the same time step, explicitly modeling physical-virtual consistency deviations:

$$\mathcal{E}_{\mathrm{E}} = \left\{ \left\langle \mathbf{v}_s^{\mathrm{RE}}(t), \mathbf{v}_s^{\mathrm{ER}}(t) \right\rangle \mid s \in \mathcal{S} \right\}_{t=1}^{K}. \quad (14)$$

The final edge set is $\mathcal{E} = \mathcal{E}_{\mathrm{T}} \cup \mathcal{E}_{\mathrm{C}} \cup \mathcal{E}_{\mathrm{E}}$, forming a physics-semantic inductive bias for graph learning. Temporal edges encode local continuity in UAV dynamics, sensor-coupling edges capture the non-independence of IMU channels under rigid-body motion, and error-correction edges explicitly link physical measurements to DT-derived residual references. Constraining message passing within these physically grounded relations reduces reliance on spurious correlations and enforces representations that preserve dynamic consistency, cross-sensor interactions, and physical-virtual deviation semantics.

F. Physics-semantic rule-based attention

Conventional self-attention primarily assigns edge weights according to feature similarity, which cannot fully characterize the heterogeneous physical relations among UAV temporal dynamics, sensor couplings, and DT-consistency interactions. To incorporate domain knowledge while preserving data-driven learning, we introduce a rule-guided edge prior as a physics-informed structural regularizer for graph propagation. The prior assigns distinct baseline strengths to temporal, sensor-coupling, and DT-consistency edges, and further modulates them using sensor stability and local covariance. Thus, the rule-guided weight is not an independent classifier, but a physics-semantic bias that steers message passing toward diagnostically meaningful interactions. To distinguish it from the GAT attention logit $s_{ij}^{(l)}$, the rule-guided edge weight is denoted as w_{ij}. Its update rule is

$$w_{ij} \leftarrow \lambda_{\kappa(i,j)} w_{ij} + \varepsilon_{ij}, \quad (15)$$

where $\kappa(i,j) \in \{\mathrm{T}, \mathrm{C}, \mathrm{E}\}$ denotes the type of edge $\langle i,j \rangle$, $\lambda_{\kappa(i,j)}$ is the edge-type coefficient, and ε_{ij} is a sensor-quality reward term. The edge-type coefficient is defined as

$$\lambda_{\kappa(i,j)} = \begin{cases} \lambda_1, & \text{if } \langle i,j \rangle \in \mathcal{E}_{\mathrm{T}}, \\ \lambda_2, & \text{if } \langle i,j \rangle \in \mathcal{E}_{\mathrm{C}}, \\ \lambda_3, & \text{if } \langle i,j \rangle \in \mathcal{E}_{\mathrm{E}}, \end{cases} \quad (16)$$

where the default configuration follows $0 < \lambda_1 < \lambda_2 < \lambda_3 < 1$ to emphasize the stronger diagnostic prior of DT-consistency edges.

Remark 3: In the default configuration, $0 < \lambda_1 < \lambda_2 < \lambda_3 < 1$ reflects the distinct diagnostic roles of the three edge types. Temporal edges preserve short-term dynamic continuity, while long-range dependencies are further captured by the BiGRU. Sensor-coupling edges encode rigid-body and IMU cross-channel interactions closely related to fault propagation. DT error-correction edges provide the strongest diagnostic prior by comparing physical responses with synchronized healthy references. Assigning larger coefficients to sensor-coupling and DT-consistency edges therefore acts as a physics-semantic regularizer that emphasizes fault-relevant interactions while preserving temporal continuity through graph propagation and recurrent modeling.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

The reward term combines a stability weight and a covariance-based decay term

$$\varepsilon_{ij} = \omega_{q,ij}\omega_{c,ij}. \tag{17}$$

For the sensors associated with nodes i and j, denoted by s_i and s_j, the stability weight is

$$\omega_{q,ij} = \frac{\max_{s\in\mathcal{S}} Q_s}{(Q_{s_i} + Q_{s_j} + \epsilon)^{\gamma}}, \tag{18}$$

with

$$Q_s = \frac{\mathrm{Std}(\mathbf{v}_s^{\mathrm{RV}})}{|\,\mathrm{Mean}(\mathbf{v}_s^{\mathrm{RV}})| + \epsilon}, \tag{19}$$

where Q_s is the coefficient of variation of sensor s, $\gamma > 0$ is a stability scaling factor, and ϵ is a small constant for numerical stability. Here $\mathbf{v}_s^{\mathrm{RV}}$ denotes the sequence of real and residual node features associated with sensor s within the current input window. The covariance-based decay term is defined as

$$\omega_{c,ij} = \exp(-\delta C_{ij}), \tag{20}$$

where $\delta > 0$ is the decay coefficient and

$$C_{ij} = \left|\mathrm{Cov}\left(\tilde{\mathbf{v}}_{s_i}^{\mathrm{RV}}, \tilde{\mathbf{v}}_{s_j}^{\mathrm{RV}}\right)\right| \tag{21}$$

measures the covariance magnitude between the normalized sensor-feature sequences. The normalized feature sequence is

$$\tilde{\mathbf{v}}_s^{\mathrm{RV}} = \frac{\mathbf{v}_s^{\mathrm{RV}} - \min(\mathbf{v}_s^{\mathrm{RV}})}{\max(\mathbf{v}_s^{\mathrm{RV}}) - \min(\mathbf{v}_s^{\mathrm{RV}}) + \epsilon}. \tag{22}$$

Combining the above definitions, the final rule-guided edge update can be written as

$$w_{ij} \leftarrow \lambda_{\kappa(i,j)} w_{ij} + \frac{\max_{s\in\mathcal{S}} Q_s}{(Q_{s_i} + Q_{s_j} + \epsilon)^{\gamma}} \exp(-\delta C_{ij}). \tag{23}$$

G. Spatiotemporal consistency learning network

Building upon the DT-consistent graph representation (Section III-B) and physics-semantic attention mechanism (Section III-F), TC-SGDN performs unified spatiotemporal consistency learning for UAV fault diagnosis. Unlike conventional architectures that separately model temporal and graph dependencies, TC-SGDN propagates information directly over DT-consistent graph representations. As shown in Fig. 4, the network combines a GAT module for spatial consistency learning with a BiGRU module for temporal dependency modeling.

The network adopts a fully symmetric architecture, where the GAT input is organized as $(K \times \text{node_type}, \text{node_features})$. Here, K denotes the temporal window length, node_type $= 6$ corresponds to accelerometer, gyroscope, and magnetometer physical/residual nodes, and node_features $= 3$ represents three-axis measurements or residuals. Each GAT head outputs 32-dimensional features; after Swish activation and four-head concatenation, the graph representation dimension becomes 128. The features are then reshaped into GRU inputs of size $(\text{Batch_size}, \text{Seq_len}, \text{Input_size})$ with Seq_len $= K$. The BiGRU uses a hidden dimension of 64 and produces 128-dimensional spatiotemporal embeddings, which are finally mapped to fault-category probabilities by the classifier.

TABLE II
DATA COLLECTION DURING THE VIR2VIR AND REA2VIR STAGES

Scenario	Vir2Vir		Rea2Vir	
	Conditions	No.	Conditions	No.
High-Maneu.	F1A1-X	1	F1A1-X	1
	F2A1-X	1	F2A1-X	1
	F2A1-Y	1	F2A1-Y	1
	F3A1-X	1	F3A1-X	1
	F3A1-Y	1	F3A1-Y	1
	F4A1-X	1	F4A1-X	1
Normal	Takeoff (C1)	10	Takeoff (C1)	3
	Hovering (C1)	10	Hovering (C1)	3
	Cruising (C1)	10	Cruising (C1)	2
Fault	Fault 1: OMPD (C4)	10	Fault 1: OMPD (C4)	10
	Fault 2: OMSJ (C3)	10	Fault 2: OMSJ (C3)	10
	Fault 3: ANI (C2)	10	Fault 3: ANI (C2)	10

Note: OMPD: One motor power drops, OMSJ: One motor signal jumps, ANI: Accelerometer noise interference. {FmAn} denotes maneuver patterns parameterized by frequency "m" and amplitude "n". C1∼C4: Tag classification. For high-maneuver rows, "No." denotes the number of maneuver configurations rather than the number of independent trials or windowed evaluation samples. Detailed fault-injection protocol and dataset construction metadata are provided in the public repository.

TABLE III
NETWORK MODEL PARAMETER CONFIGURATION

Parameter	Value	Parameter	Value
K	80	γ	1
λ_1	0.1	λ_2	0.2
λ_3	0.7	τ_{win}	10
node_type	6	*dropout*	0.3
head_num	4	*learning*	0.001
batch_size	64	*node_features*	3

Here, λ_1, λ_2, and λ_3 are the rule-guided edge-type coefficients in Eq. (16), and τ_{win} denotes the window-related training parameter used in the implementation.

H. Diagnosis classifier

The feature extraction module of the diagnostic classifier comprises a GAT and a BiGRU as illustrated in Fig. 4, which respectively extract spatial and temporal features from the data. These spatiotemporal features are then passed through a fully connected layer and a softmax layer to produce predicted probabilities for each fault category. To improve the model's generalization and prevent overfitting, dropout is incorporated. See Table III for specific parameters.

Computational complexity and deployment scale: For the default setting in Table III, each input window contains $K = 80$ time steps with 6 node types per step (three real IMU nodes and three DT-residual nodes), yielding $N_v = 6K = 480$ graph nodes. The corresponding sparse edge set contains temporal edges, sensor-coupling edges, and error-correction edges, giving approximately $N_e = 6(K-1) + 6K + 3K \approx 1.2 \times 10^3$ directed edges per window. Therefore, graph propagation remains sparse and scales linearly with the temporal window length K under a fixed sensor suite. The runtime cost of TC-SGDN mainly comes from three components: virtual-physical synchronization and residual construction, sparse multi-head GAT propagation over (N_v, N_e), and BiGRU temporal encoding over length K. Since the number of sensed channels is small and fixed in the current UAV platform, the practical inference cost remains bounded, making the framework suitable for online health monitoring.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

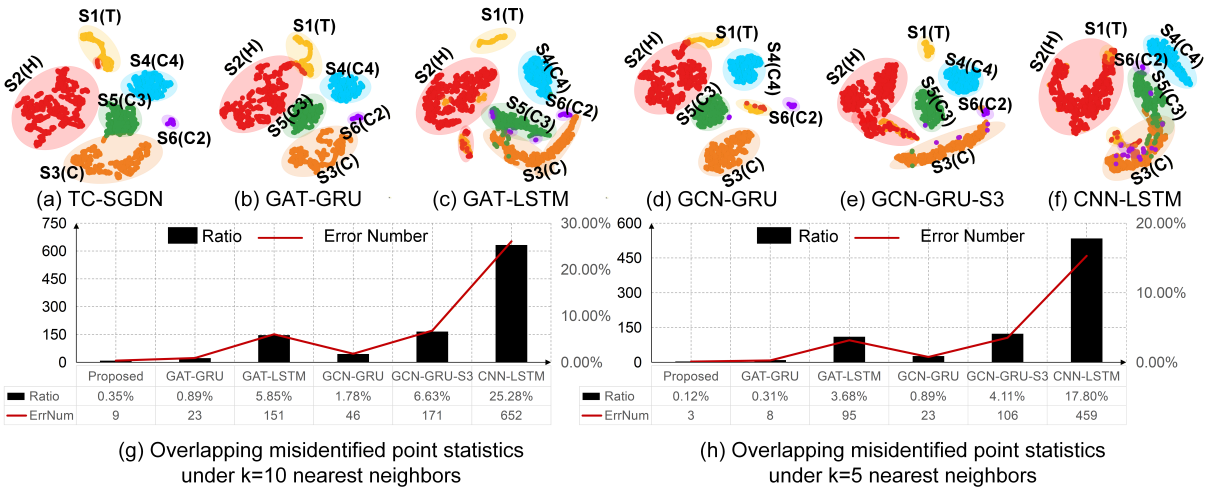


Fig. 6. t-SNE visualizations of different models. S1(T), S2(H), and S3(C) denote takeoff, hovering, and cruising, respectively, while S4(C4), S5(C3), and S6(C2) correspond to OMPD, OMSJ, and ANI faults, respectively, as listed in Table II. In panels (g) and (h), misclassified samples are identified using k-nearest neighbors with $k = 10$ and $k = 5$, respectively. "ErrNum" indicates the number of misclassifications, and "Ratio" denotes the overall misclassification rate.

IV. Experimental Study

This section evaluates the effectiveness, robustness, and generalizability of the proposed DT-based diagnostic framework through comprehensive experiments. The experimental objectives are summarized as follows.

i) ***Objective 1:*** Assess diagnostic performance under different training paradigms via two stages. *Vir2Vir* stage trains and tests the model using paired DT UAVs under identical control inputs, establishing a controlled baseline in a fully simulated setting. *Rea2Vir* stage employs paired data from a physical UAV and its DT to evaluate robustness under real-world disturbances, sensor noise, and system uncertainties. Joint analysis of both stages quantifies performance across idealized and practical scenarios.

ii) ***Objective 2:*** Evaluate diagnostic accuracy under high-maneuverability and extreme flight conditions, verifying effectiveness beyond the training distribution.

iii) ***Objective 3:*** Conduct controlled component ablation and parameter-sensitivity analyses to distinguish the contribution of each physical-semantic design from the effect of overall model complexity.

iv) ***Objective 4:*** Validate diagnostic consistency across heterogeneous datasets, UAV platforms, data distributions, and noise levels, demonstrating adaptability to diverse real-world conditions.

A. Real/virtual node architecture principle

Fig. 5(a) and Fig. 5(b) show the architectures of the real and virtual nodes. The real node comprises the physical UAV, including the airframe and autopilot hardware and software. The virtual node represents the UAV motion state through a DT, with the autopilot software executed in a virtual environment to enable real-time dynamic simulation. The corresponding real-time interaction framework is shown in Fig.2. The DT modeling framework, particularly the UAV motion model described in Section III-B1 and depicted in Fig.5(c), consists

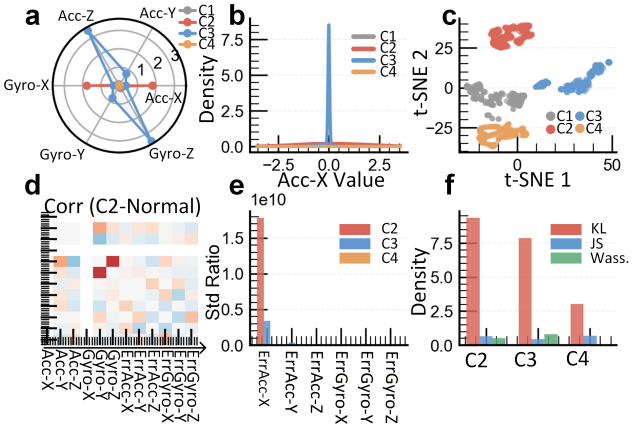


Fig. 7. Quantitative fault signature characterization. (a) Radar chart of standard deviations (Std) across six raw IMU channels. (b) Acc-x channel Kernel Density Estimation (KDE). (c) 2D t-SNE embedding. (d) Correlation difference heatmap, ΔCorr = Corr(C2)−Corr(Normal). (e) DT error response ratios (Std_fault/Std_normal) for C2–C4 (Table II). (f) Distribution metrics (KL, JS, and Wasserstein distances) between each fault class and C1 normal.

TABLE IV
Performance Comparison of Models in the Vir2Vir Stage Across Datasets

Methods	Accuracy ↑		
	Dataset 1	Dataset 2	Dataset 3
CNN-LSTM	96.91 / −0.52	95.95 / −2.84	45.72 / −53.76
GCN-GRU-S3	96.41 / −1.02	94.01 / −4.78	39.63 / −59.85
GAT-LSTM	96.67 / −0.76	95.06 / −3.73	45.31 / −54.17
GCN-GRU	97.24 / −0.19	97.82 / −0.97	66.07 / −33.41
GAT-GRU	97.31 / −0.12	97.01 / −1.78	76.15 / −23.33
FF-SAN	94.46 / −2.97	91.03 / −7.76	25.91 / −73.57
SAE-CNN	92.28 / −5.15	95.03 / −3.76	71.93 / −27.55
CNN-Transformer	97.32 / −0.11	94.04 / −4.75	66.07 / −33.41
WaveletCNN	97.37 / −0.06	96.72 / −2.07	51.62 / −47.86
Proposed	**97.43**	**98.79**	**99.48**

Dataset 1 corresponds to the Vir2Vir data in Table II. Datasets 2 and 3 are subsets of the RflyMAD public dataset with, respectively, seen and unseen training flight conditions. Values after "/" indicate accuracy differences relative to the proposed method. Gray shading denotes methods without a DT mechanism, while unshaded entries indicate DT-based approaches.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

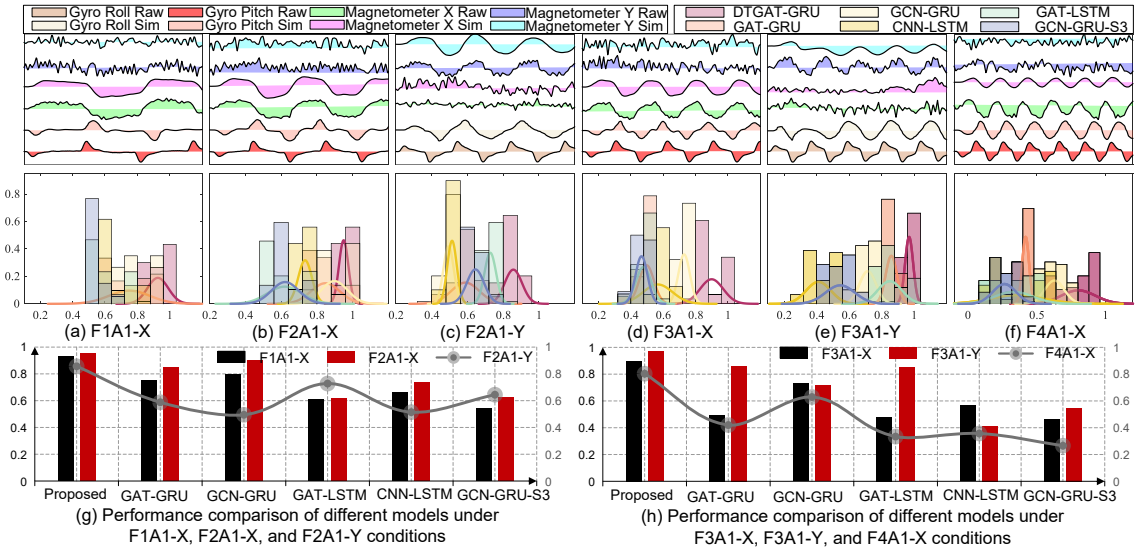


Fig. 8. Ablation results on UAV flight time-series data under varying maneuver frequencies and amplitudes, along with time-step-wise diagnostic accuracy distributions. Panels (a), (b), (d), and (f) show gyroscope and accelerometer signals at maneuver frequencies of 1–4 with amplitude 1, while the lower panels report the corresponding accuracy distributions over time. Panels (c) and (e) present time-series data and value distributions at frequencies 2 and 3. Panels (g) and (h) compare diagnostic accuracy across models under different operating conditions.

of three subsystems: airframe, sensor, and three-dimensional environment. The airframe subsystem models UAV dynamics, energy consumption, and fault behaviors, while the sensor subsystem captures electronic hardware, sensor data, and communication interfaces. The environment subsystem simulates flight scenes with obstacles and terrain, providing visual support for autonomous control. Detailed implementation is available at https://github.com/RflySim/.

B. Data collection

1) Scene Description: The FS-X150 quadcopter from Feisi Laboratory[1] (Fig.2) serves as the experimental platform, with a motion capture system providing precise position measurements. The onboard computing unit is equipped with a Rockchip RK3588[2] processor for real-time data acquisition and communication, while high-performance inference is executed on a ground station with an NVIDIA RTX 3060 GPU. The sensing system consists of a 6-axis IMU (accelerometer and gyroscope), from which both raw measurements and DT-derived residual features are constructed. To support DT operation, a co-simulation toolchain is established, including CopterSim[3] for high-fidelity dynamic modeling, RflySim3D[4] for real-time visualization and virtual-physical synchronization, and QGroundControl[5] for flight monitoring and telemetry management. Communication between the UAV and the ground station is implemented via a UDP/MAVLink protocol to ensure low-latency data transmission. To validate *Objectives 1 and 2*, flight data were collected under normal, high-maneuver, and fault conditions. Faults were injected in the real UAV by modifying the PX4 autopilot firmware to enable controllable fault triggering during flight, while corresponding faults in the virtual node were introduced by adjusting the DT model modules to maintain virtual-physical consistency.

TABLE V
PERFORMANCE COMPARISON OF MODELS IN THE REA2VIR STAGE ACROSS DATASETS

Methods	Accuracy ↑		
	Dataset 1	**Dataset 2**	**Dataset 3**
CNN-LSTM	82.22 / −15.89	97.47 / −2.38	46.42 / −53.51
GCN-GRU-S3	83.35 / −14.76	57.27 / −42.58	36.10 / −63.83
GAT-LSTM	76.05 / −22.06	69.02 / −30.83	26.75 / −73.18
GCN-GRU	91.60 / −6.51	99.66 / −0.19	98.68 / −1.25
GAT-GRU	91.79 / −6.32	99.67 / −0.18	98.11 / −1.82
FF-SAN	83.75 / −14.36	90.62 / −9.23	52.80 / −47.13
SAE-CNN	71.46 / −26.65	99.91 / +0.06	98.43 / −1.50
CNN-Transformer	76.78 / −21.33	99.24 / −0.61	95.60 / −4.33
WaveletCNN	91.85 / −6.26	1.00 / +0.15	99.90 / −0.03
Proposed	**98.11**	**99.85**	**99.93**

Dataset 1 corresponds to the Vir2Vir data in Table II. Datasets 2 and 3 are subsets of the RflyMAD public dataset with, respectively, seen and unseen training flight conditions. Values after "/" indicate accuracy differences relative to the proposed method. Gray shading denotes methods without a DT mechanism, while unshaded entries indicate DT-based approaches.

Remark 4: High-maneuver data covering sharp turns, emergency braking, and other aggressive maneuvers were used exclusively for validation to assess performance under high-dynamic conditions. Data from normal and fault scenarios were used for training, preventing misclassification caused by abrupt sensor fluctuations and enabling rigorous evaluation of robustness and adaptability to complex flight environments.

2) Dataset Architecture: Data collection follows a two-stage design: *Vir2Vir* validation and *Rea2Vir* deployment. In the *Vir2Vir* stage, both healthy and faulty nodes are generated by identical DT models under the same control inputs. In the *Rea2Vir* stage, faulty virtual nodes are replaced with real UAV data. To address Objective 4, three datasets are used to evaluate performance across diverse conditions and data sources:

[1]Lab Intro: http://feisilab.com/
[2]Hardware details: https://www.rock-chips.com/a/en/products/RK35_Series/2022/0926/1660.html
[3]CopterSim repository: https://github.com/RflySim/CopterSim
[4]RflySim3D documentation: https://github.com/RflySim/Docs
[5]QGroundControl: https://qgroundcontrol.com/

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

TABLE VI
PERFORMANCE COMPARISON OF MODELS IN THE REA2VIR STAGE UNDER DIFFERENT MANEUVERING CONDITIONS

Methods	Accuracy ↑					
	F1A1-X	*F2A1-X*	*F3A1-X*	*F4A1-X*	*F2A1-Y*	*F3A1-Y*
CNN-LSTM	0.8798 / −0.1059	0.7252 / −0.2578	0.5554 / −0.4314	0.5169 / −0.3950	0.6112 / −0.3216	0.5344 / −0.4656
GCN-GRU-S3	0.8398 / −0.1459	0.6811 / −0.3019	0.5714 / −0.4154	0.5430 / −0.3689	0.6577 / −0.2751	0.5971 / −0.4029
GAT-LSTM	0.8208 / −0.1649	0.6608 / −0.3222	0.5247 / −0.4621	0.4969 / −0.4150	0.8345 / −0.0983	0.9687 / −0.0313
GCN-GRU	0.9613 / −0.0244	0.9681 / −0.0149	0.8439 / −0.1429	0.8004 / −0.1115	0.7260 / −0.2068	0.8580 / −0.1420
GAT-GRU	0.9627 / −0.0230	0.9579 / −0.0251	0.6509 / −0.3359	0.6875 / −0.2244	0.8242 / −0.1086	0.9415 / −0.0585
FF-SAN	0.8004 / −0.1853	0.5746 / −0.4084	0.4369 / −0.5499	0.4714 / −0.4405	0.5377 / −0.3951	0.4760 / −0.5240
SAE-CNN	0.9450 / −0.0407	0.8704 / −0.1126	0.7157 / −0.2711	0.6999 / −0.2120	0.7590 / −0.1738	0.9186 / −0.0814
CNN-Transformer	0.9722 / −0.0135	0.8874 / −0.0956	0.5875 / −0.3993	0.5079 / −0.4040	0.7394 / −0.1988	0.9040 / −0.0960
WaveletCNN	0.8086 / −0.1771	0.6133 / −0.3697	0.4815 / −0.5053	0.5120 / −0.3999	0.4364 / −0.4964	0.4384 / −0.5616
Proposed	**0.9857**	**0.9830**	**0.9868**	**0.9119**	**0.9328**	**1.0000**

{FmAn} denotes maneuver patterns parameterized by frequency "m" and amplitude "n", used to simulate diverse flight conditions, where X and Y indicate maneuvers applied along the corresponding coordinate axes. Values after "/" indicate accuracy differences relative to the proposed method. Gray shading denotes methods without a DT mechanism, while unshaded entries indicate DT-based approaches.

i) ***Dataset 1: Custom real-flight dataset***. Collected from 110 real UAV missions via the DT platform (Fig. 2), covering takeoff, hovering, and cruising, with both normal and faulty cases (Table II). Used for training and initial validation.

ii) ***Dataset 2: RflyMAD (matched conditions)*** [41]. Used to assess generalization under aligned flight conditions. Although DT-free, simulation and real data are treated as virtual and physical nodes, respectively, with inputs constructed via the same graph embedding.

iii) ***Dataset 3: RflyMAD (unseen conditions)*** [41]. Includes unseen flight scenarios; the same graph construction enables evaluation of robustness and generalization under distribution shifts.

To enhance interpretability and reproducibility, we establish a unified modeling framework covering the physical origins, injection mechanisms, and statistical characteristics of three UAV fault types shown in Table II. By modifying the PX4 firmware, a controllable trigger-based online fault injection interface is implemented for real-time in-flight fault activation. Specifically, efficiency degradation (C4) is simulated via scaling factors applied to control and actuation outputs; sensor anomalies (C2) and motor perturbations (C3) are injected as Gaussian white noise into the accelerometer and multi-channel gyroscope, representing single-channel and coupled multi-channel disturbances, respectively.

Remark 5: Rather than arbitrary noise, the injected faults emulate practical UAV degradations: additive sensor inconsistency (C2: acceleration interference), coupled dynamic disturbances (C3: actuator-body perturbation), and multiplicative efficiency loss (C4: actuator scaling). Analyzing joint raw-DT residual features (Fig. 7) reveals their distinct signatures. C2 induces pronounced single-channel shifts for high separability. C3 variations, while concealed in raw space, are significantly amplified by DT residuals, validating enhanced sensitivity to coupled anomalies. Conversely, C4 heavily overlaps with normal states in both spaces, exposing the fundamental inability of additive residuals to capture multiplicative degradation.

C. Vir2Vir verification

1) Comparison with state-of-the-art: To validate *Objectives 1 and 4*, this paper selects the CNN-LSTM architecture used in [29] and the GAT-GRU structure from [30], both of which are representative UAV fault-diagnosis networks. GCN-GRU, GCN-GRU-S3, and GAT-LSTM are also included to separate the effects of graph modeling, temporal modeling, and DT-derived inputs. Specifically, TC-SGDN, GAT-GRU, and GCN-GRU use paired physical/virtual node information, whereas CNN-LSTM, GCN-GRU-S3, and GAT-LSTM rely on single-source sensor inputs. This comparison is designed not only to rank model accuracy, but also to examine whether the DT-derived health reference changes the diagnostic behavior under distribution shifts.

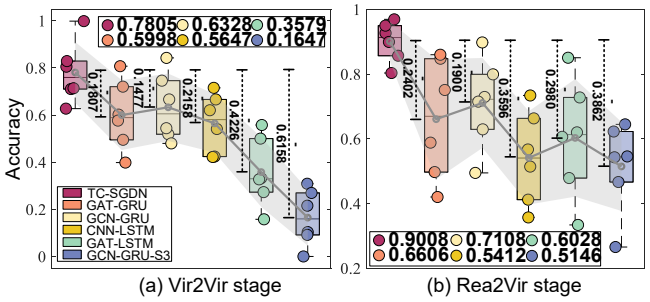


Fig. 9. Boxplots illustrate diagnostic accuracy distributions for the *Vir2Vir* and *Rea2Vir* stages under varying maneuver frequencies and amplitudes. They reveal accuracy variability across models and conditions, with circular markers indicating mean accuracy. Dashed lines highlight performance gaps relative to the proposed TC-SGDN, emphasizing its diagnostic superiority.

Table IV reports the diagnostic results in the *Vir2Vir* stage. Under Dataset 1 and Dataset 2, several temporal or graph-based baselines already achieve high accuracy, indicating that when the training and testing conditions are relatively aligned, backbone capacity alone can explain much of the performance. The more informative case is Dataset 3, which contains unseen operating conditions. In this setting, single-source baselines degrade sharply, while TC-SGDN maintains 99.48% accuracy. This contrast suggests that the main advantage of the proposed method is not merely higher fitting capacity, but the use of a synchronized DT reference to convert raw sensor-pattern classification into consistency-based deviation modeling. The comparison with GCN-GRU and GAT-GRU further shows that DT information alone is insufficient: without the proposed twin-consistent graph construction and temporal consistency learning, the models still struggle to preserve discriminability

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

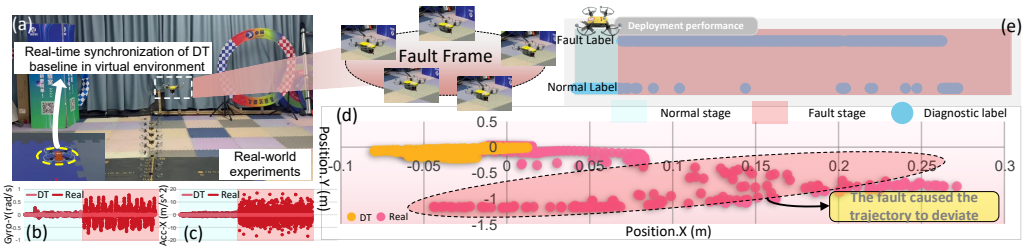


Fig. 10. Real-world UAV experiments. (a) DT baseline real-time synchronization and experimental setup. Panels (b) and (c) compare DT and real UAV gyroscope/accelerometer data. (d) Fault-induced trajectory deviation between DT and real data. (e) Diagnostic normal/fault stages and deployment performance.

TABLE VII
COMPREHENSIVE REAL-TIME DEPLOYMENT PERFORMANCE OF THE DT-DRIVEN FAULT DIAGNOSIS SYSTEM

Model Inference	**Resource Consumption**	**End-to-End Latency**	**Sensor Scalability**
P50 (Steady): 10.13 ms	Total CPU: ∼232%	IMU Sampling (50 Hz): ∼20 ms	6 Sensors (Curr): ∼10 ms
Mean (w/ Jitter): 38.15 ms	Total Memory: ∼3.1 GB	DT Sync + Feature: ∼11–21 ms	9 Sensors (w/ Mag): ∼15 ms
P95 (Worst-case): 126.39 ms	GPU VRAM: 3.35/6 GB (54.5%)	Model Inference (P50): ∼10 ms	12 Sensors (Redun): ∼20 ms
Throughput: 143.76 Hz	GPU Util: 23.5% avg (peak 90%)	Total: 43–54 ms (19–23 Hz)	(All real-time capable)

Note: Model inference latency refers only to TC-SGDN forward propagation, whereas end-to-end latency includes IMU sampling, DT synchronization, residual-feature construction, and inference. The achieved 19–23 Hz online diagnostic rate is intended for outer-loop health monitoring and fault response. Under the current default setting (K=80, 6 node types, 3 features per node), the graph size is fixed at 480 nodes per window, which keeps the diagnosis module computationally bounded for the present UAV sensing configuration.

TABLE VIII
ABLATION STUDY

Components	Acc ↑	F1 ↑	Rec ↑	Prec ↑
w/ All	**0.981**	**0.928**	**0.920**	**0.937**
w/o RAT	0.977 / 0.004	0.917 / 0.011	0.919 / 0.001	0.915 / 0.022
w/o ECE	0.969 / 0.012	0.918 / 0.010	0.916 / 0.004	0.920 / 0.017
w/o SCE	0.968 / 0.013	0.914 / 0.014	0.919 / 0.001	0.909 / 0.028
w/o EPO	0.938 / 0.043	0.871 / 0.057	0.908 / 0.012	0.836 / 0.101
w/o TDE	0.831 / 0.150	0.731 / 0.197	0.769 / 0.151	0.697 / 0.240

Note: Gray values denote performance degradation vs. the full module (w/ All); red bold values indicate the most significant degradation. The "w/o" modules from top to bottom correspond to equations (16), (14), (13), (19-22) and (12) respectively. All ablation variants share identical training settings and backbone structures; only the target physical-semantic component is removed or neutralized (RAT, ECE, SCE, EPO, or TDE), ensuring that performance differences arise solely from the excluded design element.

TABLE IX
SENSITIVITY ANALYSIS OF RULE-GUIDED EDGE COEFFICIENTS (CORRESPONDING TO EQUATIONS (16) AND (18))

Parameter setting	**Acc. ↑**	**F1 ↑**	**Precision ↑**
Default: $(\lambda_1, \lambda_2, \lambda_3) = (0.1, 0.2, 0.7)$, $\gamma = 1$	0.981	0.928	0.937
Uniform edge prior: $(0.33, 0.33, 0.33)$	0.979	0.933	0.947
Weak DT prior: $(0.5, 0.3, 0.2)$	0.980	0.934	0.949
Strong DT prior: $(0.05, 0.15, 0.8)$	0.973	0.912	0.910
$\gamma = 0.2$	0.986	0.964	0.982
$\gamma = 2.0$	0.976	0.933	0.946

under unseen flight dynamics.

D. Rea2Vir verification

1) Comparison with state-of-the-art: Following the *Vir2Vir* validation, the *Rea2Vir* stage examines whether the learned physical-virtual consistency remains effective when real UAV measurements contain disturbances, sensor noise, and modeling mismatch. As shown in Fig. 6, TC-SGDN produces compact and separable latent clusters, while several baselines exhibit partial feature overlap across flight states and fault classes. This observation is important because feature overlap is precisely the failure mode expected under high-dynamic UAV maneuvers: normal aggressive responses and fault-induced deviations can occupy similar raw-signal regions. By embedding DT residual nodes and physics-semantic edges into the graph, TC-SGDN makes the latent representation more aligned with deviations from the expected healthy response rather than with raw signal amplitude alone.

Table V further shows that TC-SGDN achieves the most stable overall performance across the custom real-flight data and the two RflyMAD subsets. Although some baselines are competitive on individual public-data subsets, their performance varies substantially across datasets, especially on Dataset 1 and Dataset 3. This indicates that the key benefit of the proposed framework is cross-condition stability: the DT baseline provides a common reference for heterogeneous data sources, while the graph-temporal structure preserves sensor coupling and temporal continuity during real-to-virtual transfer.

2) Ablation study: To quantify the contribution of each module in the proposed TC-SGDN and verify the framework design, ablation experiments were conducted by sequentially removing key components (RAT, ECE, SCE, EPO, and TDE) from the full model. Performance degradation across Accuracy, F1-score, Recall, and Precision indicates the contribution ranking: TDE > EPO > SCE > ECE > RAT. Notably, removing TDE causes the most severe degradation, confirming that modeling temporal dynamic correlations is essential for distinguishing high-maneuver normal behaviors from fault patterns. Eliminating EPO leads to a marked Precision drop, demonstrating that explicitly modeling sensor coupling strength via covariance-based weighting effectively suppresses noise-induced misdiagnosis and enhances spatial semantic representation. In contrast, removing SCE, ECE, or RAT results in only minor performance changes, indicating their auxiliary optimization roles: SCE enriches same-step

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

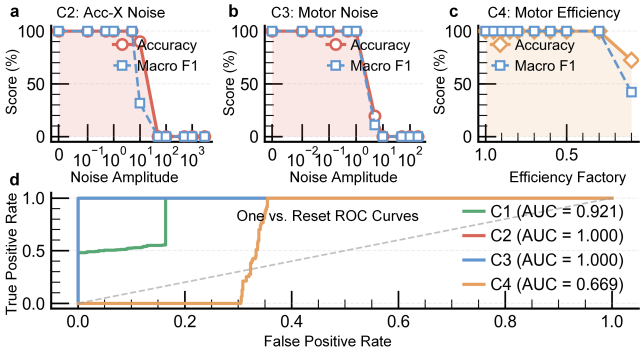


Fig. 11. Diagnostic performance degradation under controlled fault-emulation severities. Faults are injected into normal base samples via additive sensor disturbances $x' = x + \mathcal{N}(0, A_{\text{noise}})$ or multiplicative actuator scaling $x' = x \times \eta_{\text{eff}}$, where A_{noise} and η_{eff} represent the disturbance amplitude and efficiency factor, respectively. (a) C2: Additive disturbance in Acc-x. (b) C3: Signal-jump applied to Motor #3. (c) C4: Multiplicative scaling η_{eff} on Motor #3. (d) One-vs-Rest Receiver Operating Characteristic (ROC) curves on the baseline test set. The ROC plots True Positive Rate (TPR $=$ TP$/$(TP $+$ FN)) against False Positive Rate (FPR $=$ FP$/$(FP $+$ TN)) across thresholds; the Area Under the Curve (AUC) quantifies overall discriminability (AUC $= 1.0$ for perfect separation, AUC $= 0.5$ for random guessing).

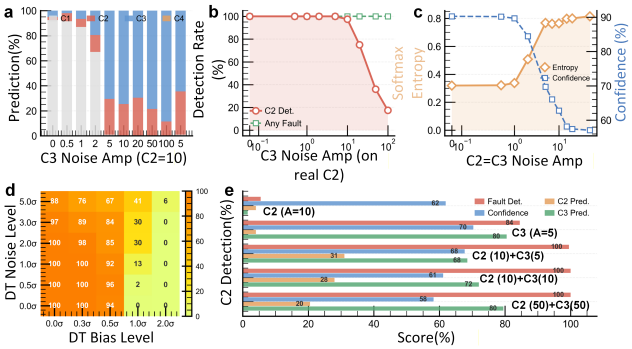


Fig. 12. Model response characteristics under simultaneous multi-fault injection. (a) Stacked bar chart of prediction distribution (%) when C2 noise is fixed at 10 m/s^2 and C3 noise amplitude sweeps from 0 to 100. (b) Fault interference: C2 detection rate and any-fault detection rate on real C2 fault data with increasing C3 noise superimposed. (c) Dual-axis plot of softmax entropy and maximum prediction confidence under equal-strength C2=C3 noise injection. (d) Noise $\times$ Bias heatmap: C2 detection rate (%) on real C2 data with DT-channel noise (rows: 0–5σ) and systematic bias (columns: 0–2σ) perturbation. (e) Horizontal grouped bar chart comparing five scenarios.

sensor coupling, ECE compensates for physical-virtual deviations, and RAT refines attention allocation based on sensor stability. Overall, the ablation results validate the hierarchical design of TC-SGDN, where TDE and EPO constitute the core spatiotemporal modeling backbone, while SCE, ECE, and RAT enhance robustness, strongly supporting the framework's physical-semantic spatiotemporal modeling and DT-driven health baseline design.

3) Sensitivity study: Table IX further examines whether the rule-guided coefficients are fragile manually tuned heuristics. The results show that TC-SGDN remains stable under several coefficient configurations: uniform edge priors and weakened DT-consistency priors achieve performance comparable to the default setting, indicating that the model does not rely on a single hand-crafted coefficient hierarchy. The strong DT-prior setting leads to lower F1-score and Precision, suggesting that over-emphasizing DT-consistency edges may disturb the balance among temporal continuity, sensor coupling, and physical-virtual residual semantics. Varying the stability scaling factor γ also does not cause performance collapse; $\gamma = 0.2$ even improves the metrics, showing that the default $\gamma = 1$ is a conservative physics-informed setting rather than an over-tuned optimum. Overall, the sensitivity analysis supports the interpretation that the coefficients act as robust structural priors for graph message passing, while the main performance gain comes from the DT-consistent graph design and temporal dependency modeling.

4) Comparison with high-maneuver scenarios: To validate *Objectives 2 and 3*, generalization tests were conducted on six *Rea2Vir* high-maneuver settings, namely F1A1-X, F2A1-X, F3A1-X, F4A1-X, F2A1-Y, and F3A1-Y, as shown in Table II and Fig. 8. The key observation is that maneuver intensity changes the raw IMU distribution in a way that can resemble fault signatures. As frequency and amplitude increase, models relying primarily on raw temporal or spatial patterns show larger fluctuations, whereas TC-SGDN maintains an average accuracy of 0.9667 and a worst-case accuracy of 0.9119 across the six high-maneuver settings in Table VI. This robustness is explained by the DT-consistent representation: the synchronized virtual node absorbs the expected maneuver-induced response, while residual and coupling edges emphasize deviations that are less consistent with healthy dynamics.

Table VI and Fig. 9 therefore provide a stronger insight than a simple performance ranking. The improvement is most meaningful under aggressive maneuvers, where the diagnostic problem is no longer only classification among known patterns but discrimination between normal high-dynamic behavior and true abnormal deviations. The remaining performance drop in the most difficult cases, such as F4A1-X and F2A1-Y, also indicates that extremely strong maneuvers can still distort the residual baseline, motivating the robustness analysis in the following discussion. This observation is also central to the accuracy-cost trade-off. The added DT synchronization and graph-temporal computation are most valuable when the operating condition is difficult enough that raw-signal models lose discriminability. In the six high-maneuver settings, TC-SGDN achieves an average accuracy of 0.9667 and a worst-case accuracy of 0.9119, whereas several lighter baselines suffer large drops under the same conditions. Therefore, the additional computational cost is not primarily justified by marginal gains in easy matched cases, but by the improved robustness and worst-case stability under high-dynamic conditions.

E. Real-world deployment experiment

As shown in Fig. 10, the real-world experiment provides evidence that the DT can operate as an online diagnostic reference rather than an offline simulation tool. During normal hovering, the DT and real UAV measurements remain dynamically aligned, so the residual graph stays close to the healthy baseline. After fault occurrence, the real trajectory gradually deviates from the DT response, and this loss of physical-virtual consistency becomes an interpretable diagnostic cue.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

TABLE X
ROBUSTNESS TEST UNDER PARAMETER PERTURBATION, SYNCHRONIZATION ERROR, AND DATA QUALITY DEGRADATION

Panel A: Perturbation Sens.					Panel B: Sync. Rob.				Panel C: Data Qual. Impact				
Pert. Type	**Level**	**Acc.(%)**	**F1(%)**	**Deg.(%)**	**Err. Type**	**Test Level**	**Acc.(%)**	**Deg.(%)**	**Factor**	**Test Level**	**Acc.(%)**	**F1(%)**	**Deg.(%)**
Gauss. Noise	0.5σ	93.70	73.43	-0.37	Cross-Mode	30% Mism.	92.71	-1.36	SNR Deg.	5dB	94.19	73.92	0.12
	1.0σ	89.25	69.17	-4.82		50% Mism.	90.98	-3.09		0dB	92.46	71.96	-1.61
	1.5σ	64.89	50.14	-29.17		100% Mism.	64.52	-29.55		-5dB	30.90	33.07	-63.17
Sys. Bias	0.5σ	30.90	33.53	-63.17	Time Jit.	±100 Samp.	91.22	-2.85	DT Corr.	$1.0\times\sigma$	27.32	20.22	-66.75
	1.0σ	16.19	9.86	-83.88		±300 Samp.	86.90	-7.17		$10.0\times\sigma$	15.08	11.51	-83.99
Chan.-Select.	Acc	21.01	21.76	-73.06		±700 Samp.	83.93	-10.14	Per-Class F1	Full Model	95.60	—	—
	Gyro	76.89	60.79	-17.18	Resamp. Dev.	0.5× Down	93.82	-0.25		Corrupted Model	24.63	—	—
	Mag	75.40	58.96	-18.67		0.8×–2.0×	94.07	0		—	—	—	—

Abbreviations: Gauss.=Gaussian; Sys.=Systematic; Chan.-Select.=Channel-Selective; Pert.=Perturbation; Sens.=Sensitivity; Acc.=Accuracy; Deg.=Degradation; Err.=Error; Sync.=Synchronization; Rob.=Robustness; Jit.=Jitter; Resamp.=Resampling; Dev.=Deviation; Mism.=Mismatch; Samp.=Samples; Qual.=Quality; SNR Deg.=SNR Degradation; DT Corr.=DT Corruption.

Experimental Setup: Tested under hover/forward flight with 6 IMU/Gyro channels (Mag channel excluded for NRMSE>0.88); Diagnosis model: TC-SGDN for 4-class classification (C1: Normal, C2-C4: Faults), baseline accuracy=94.07%; Sensitivity ranking: Systematic bias > Gaussian noise > cross-mode mixing > time jitter > resampling deviation.

Key Thresholds: Safe boundary (degradation<5%): Gaussian noise $<0.5\sigma$; 0.5σ systematic bias causes 63% accuracy drop, Acc channels are core diagnostic features; Model is robust to sampling rate changes and small jitter, stable below 50% cross-mode mismatch; Performance is fully stable when SNR$\geq$5dB (0dB is critical point).

Metric Formulas: NRMSE $= \frac{\text{RMSE}}{\max(y)-\min(y)}$, RMSE $= \sqrt{\frac{1}{M}\sum_{r=1}^{M}(y_r-\hat{y}_r)^2}$, MAE $= \frac{1}{M}\sum_{r=1}^{M}|y_r-\hat{y}_r|$; Accuracy $= \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$, Macro-F1 $= \frac{1}{C}\sum_{c=1}^{C}\frac{2\cdot\text{Precision}_c\cdot\text{Recall}_c}{\text{Precision}_c+\text{Recall}_c}$; SNR (dB) $= 10\log_{10}\left(\frac{\text{Signal Power}}{\text{Noise Power}}\right)$, where M denotes the number of evaluated samples and C denotes the number of classes.

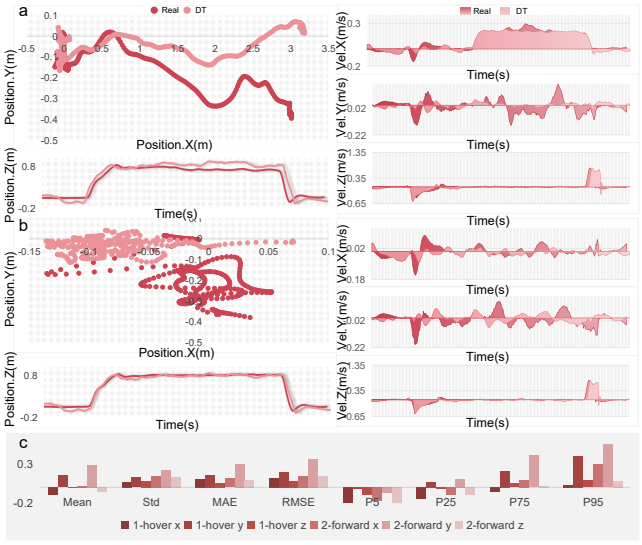


Fig. 13. Trajectory tracking accuracy of UAV DT system. (a) Hover flight; (b) Forward flight; (c) Position error statistics.

This result clarifies why the method works in deployment: the classifier does not rely only on absolute signal values, but on whether the physical UAV remains consistent with the expected healthy evolution generated by the synchronized DT.

Table VII summarizes the deployed system's computational performance. Although the diagnosis network achieves a median inference latency of 10.13 ms (143.76 Hz throughput), the complete pipeline including IMU sampling, DT synchronization, feature construction, and inference requires 43–54 ms, corresponding to a sustained online diagnostic rate of 19–23 Hz. This rate is sufficient for real-time health monitoring, fault awareness, and outer-loop fault-response support, though not intended for high-rate inner-loop flight control. The deployment results further reveal that the primary computational overhead arises from DT synchronization, residual-feature construction, and scheduling jitter in the real–virtual communication loop rather than network inference itself, explaining the gap between raw model throughput and end-to-end performance. Under the current six-sensor configuration, the graph remains fixed and sparse, enabling predictable graph-temporal computation and placing greater emphasis on stable DT communication and moderate edge-computing resources than extreme onboard processing capability. Overall, TC-SGDN is well suited for moderate-rate online diagnosis in safety-critical and highly dynamic scenarios, where robustness against maneuver-induced ambiguity outweighs minimal computational overhead. For nominal operating conditions or resource-constrained inner-loop applications, lighter diagnostic models may remain preferable.

F. Discussion

1) Fault characteristics: The fault-severity results in Fig. 11 show a step-like collapse rather than a smooth linear degradation. This suggests that TC-SGDN behaves primarily as a consistency discriminator instead of a continuous fault-severity estimator: performance remains stable while the residual pattern lies inside the learned healthy/fault-consistent region, but drops quickly once the perturbation pushes samples into an ambiguous region. The strong separability of C2 and C3 can be explained by their additive or signal-jump disturbance nature, which is directly reflected in DT residual nodes and sensor-coupling edges. In contrast, C4 is harder because its protocol uses multiplicative actuator-efficiency scaling ($\eta_{\text{eff}} = 0.85$ for Motor #3 in the benchmark). Such degradation may preserve the temporal shape of nominal dynamics while reducing the effective actuation response, producing weaker residual contrast and larger overlap with normal conditions. This explains why C4 is retained as a challenging case rather than treated as a simple injected-noise scenario.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

The multi-fault results in Fig. 12 further reveal a diagnostic boundary. When C2 and C3 are superimposed, anomaly detection can remain reliable, but fine-grained class attribution becomes unstable because the dominant residual channel can mask weaker fault signatures. The increase in entropy and the decrease in maximum confidence are therefore not only signs of performance degradation, but also useful uncertainty cues. More importantly, the sharp sensitivity to systematic bias shows that the proposed method depends on a bias-free dynamic health reference. If the DT baseline is globally shifted, the residual graph no longer represents fault-induced deviation alone, and the physical-virtual consistency semantics are corrupted.

2) Model robustness: Fig. 13 and Table X summarize how DT uncertainty affects diagnosis. The results show that the framework is relatively tolerant to stochastic perturbations, moderate synchronization mismatch, and sampling-rate changes, but highly sensitive to systematic bias and severe DT corruption. This distinction is central to understanding the method. Random noise and local timing errors are partly absorbed by temporal aggregation and graph-level coupling, so they do not necessarily destroy the residual semantics. Systematic bias, however, shifts the healthy reference itself and contaminates all residual nodes in a consistent direction, making normal behavior appear abnormal or masking true faults. The channel-selective results further show that accelerometer-related residuals are structurally important for the current fault set, because C2 directly perturbs acceleration and C3/C4 also affect translational dynamics through actuator-body coupling.

Overall, the experiments indicate that TC-SGDN does not require a perfectly accurate DT trajectory at every instant; it requires a dynamically consistent and low-bias health baseline. This is the main reason why the method can tolerate moderate noise and synchronization uncertainty while failing under systematic twin bias. The practical implication is that future deployment should prioritize online DT bias calibration and uncertainty-aware residual weighting, rather than only increasing classifier complexity.

3) Limitations and practical considerations: The proposed framework still has several practical limitations. First, TC-SGDN relies on an accurate DT to provide a healthy reference. Modeling errors, parameter drift, or unmodeled environmental effects may bias the DT baseline, causing residual graphs to capture both fault-induced and model-induced deviations and thereby reducing diagnostic reliability. Second, although the synchronization mechanism tolerates bounded timing mismatch, significant communication delays, packet disorder, or persistent asynchronous sampling can introduce synchronization artifacts into residual features. Third, the current implementation is suitable for moderate-rate outer-loop health monitoring rather than high-rate inner-loop flight control. On resource-constrained UAV platforms, the combined overhead of DT synchronization, residual-feature construction, and graph-temporal inference may still require edge assistance, model compression, or asynchronous deployment. These limitations motivate future research on DT self-calibration, uncertainty-aware residual weighting, robust synchronization under communication disturbances, and lightweight fully onboard deployment.

V. Conclusion

This work proposes a DT-guided spatiotemporal diagnosis framework that shifts UAV fault diagnosis from data-driven pattern fitting to dynamic health reference construction. By embedding a continuously synchronized DT as an internal prior, the method establishes an evolving health baseline, enabling consistent comparison between physical observations and virtual healthy counterparts. This mitigates ambiguity between normal variations and fault-induced anomalies, which is a key limitation of existing methods. Beyond performance gains, integrating DT consistency into graph-based modeling yields a structurally interpretable and deployment-ready paradigm. Extensive evaluations on virtual, real, and public datasets demonstrate stable performance across varying conditions, while real-world experiments confirm that the DT can operate online under weak synchronization without intrusive control modifications, supporting practical deployment.

These results also highlight key future directions. Although robust to stochastic disturbances and moderate misalignment, the framework is sensitive to structural deviations (e.g., systematic twin bias), motivating bias-aware modeling and online calibration to preserve residual interpretability. Moreover, synchronization-aware learning, rather than strict point-wise alignment, offers a more practical path to handle temporal and communication uncertainties. Maintaining dynamic consistency under evolving conditions and model drift thus emerges as a central challenge for scalable DT-driven diagnosis. Overall, embedding DTs as online diagnostic references rather than offline simulators provides a promising path toward reliable and generalizable UAV fault diagnosis. Future work will focus on improving twin adaptability and self-calibration, enhancing robustness to bias and domain shifts, and optimizing computational efficiency for large-scale real-time deployment.

References

[1] D. Caballero-Martin, J. M. Lopez-Guede, J. Estevez, and M. Graña. Artificial intelligence applied to drone control: A state of the art. *Drones*, 8(7):296, 2024.

[2] J. P. Li, K. Yue, Z. Q. Wu, F. Jiang, Z. Zhong, S. H. Zhang, and W. H. Li. Mtsnn: A few-shot fine-grained diagnosis framework for cross-machine fault diagnosis with heterogeneous fault categories. *IEEE Transactions on Instrumentation and Measurement*, 74:1–11, 2025.

[3] K. K. Huang, S. J. Wu, B. Sun, C. H. Yang, and W. H. Gui. Metric learning-based fault diagnosis and anomaly detection for industrial data with intraclass variance. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):547–558, 2024.

[4] J. P. Bai, J. H. Luo, Y. Chen, Y. M. Tang, and et al. The DDPG-based joint optimization of task offloading and content caching in UAV-assisted IoV. *IEEE Internet of Things Journal*, 12(19):40330–40346, 2025.

[5] C. J. Li, H. Y. Wang, Y. Z. Zhang, and et al. Coupled noise-aware UAV fault diagnosis based on learnable wavelet packet transform and scale-graph label enhancement. *IEEE Internet of Things Journal*, 12(24):52913–52926, 2025.

[6] G. K. Fourlas and G. C. Karras. A survey on fault diagnosis methods for UAVs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 394–403, 2021.

[7] L. A. Al-Haddad and A. A. Jaber. An intelligent fault diagnosis approach for multirotor UAVs based on deep neural network of multi-resolution transform features. *Drones*, 7(2):82, 2023.

[8] K. Guo, N. Wang, D. T. Liu, and X. Y. Peng. Uncertainty-aware LSTM based dynamic flight fault detection for UAV actuator. *IEEE Transactions on Instrumentation and Measurement*, 72:1–13, 2022.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2026.3709310

[9] T. Huang, Q. Zhang, X. A. Tang, S. Y. Zhao, and X. N. Lu. A novel fault diagnosis method based on CNN and LSTM and its application in fault diagnosis for complex systems. *Artificial Intelligence Review*, 55(2):1289–1315, Feb 2022.

[10] X. H. Chen, B. K. Zhang, and D. Gao. Bearing fault diagnosis base on multi-scale CNN and LSTM model. *Journal of Intelligent Manufacturing*, 32(4):971–987, Apr 2021.

[11] A. Khorram, M. Khalooei, and M. Rezghi. End-to-End CNN + LSTM deep learning approach for bearing fault diagnosis. *Applied Intelligence*, 51(2):736–751, Feb 2021.

[12] R. Y. Huang, J. P. Li, Y. X. Liao, J. B. Chen, Z. Wang, and W. H. Li. Deep adversarial capsule network for compound fault diagnosis of machinery toward multidomain generalization task. *IEEE Transactions on Instrumentation and Measurement*, 70:1–11, 2021.

[13] W. H. Li, H. Lan, J. B. Chen, K. Feng, and R. Y. Huang. WavCapsNet: An interpretable intelligent compound fault diagnosis method by backward tracking. *IEEE Transactions on Instrumentation and Measurement*, 72:1–11, 2023.

[14] R. Y. Huang, Y. Chen, and C. Liu. Revealing state-feature dependencies in dynamic degradation: An exofeature-aware transformer for battery state-of-health prediction. *IEEE Transactions on Consumer Electronics*, 72(1):719–730, 2026.

[15] D. Y. Chen, Z. X. Xie, R. N. Liu, W. L. Yu, Q. H. Hu, X. L. Li, and S. X. Ding. Bayesian hierarchical graph neural networks with uncertainty feedback for trustworthy fault diagnosis of industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):18635–18648, 2024.

[16] G. Yang, H. Tao, T. T. Yu, R. X. Du, and Y. Zhong. Online fault diagnosis of harmonic drives using semisupervised contrastive graph generative network via multimodal data. *IEEE Transactions on Industrial Electronics*, 71(3):3055–3063, 2023.

[17] Z. H. Mao, H. Wang, B. Jiang, J. Xu, and H. F. Guo. Graph convolutional neural network for intelligent fault diagnosis of machines via knowledge graph. *IEEE Transactions on Industrial Informatics*, 20(5):7862–7870, 2024.

[18] H. Wang, R. N. Liu, S. X. Ding, Q. H. Hu, Z. X. Li, and H. K. Zhou. Causal-trivial attention graph neural network for fault diagnosis of complex industrial processes. *IEEE Transactions on Industrial Informatics*, 20(2):1987–1996, 2024.

[19] K. B. Zhou, C. Y. Yang, J. Liu, and Q. Xu. Dynamic graph-based feature learning with few edges considering noisy samples for rotating machinery fault diagnosis. *IEEE Transactions on Industrial Electronics*, 69(10):10595–10604, 2021.

[20] D. Y. Chen, R. N. Liu, Q. H. Hu, and S. X. Ding. Interaction-aware graph neural networks for fault diagnosis of complex industrial processes. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):6015–6028, 2023.

[21] W. J. Lang, Y. H. Hu, Q. F. Li, H. Q. Wen, and Y. B. Salamah. Inductive graph neural network for virtual vibration sensor reconstruction in PMSM powertrain. *IEEE Transactions on Industrial Electronics*, 71(10):13288–13298, 2024.

[22] X. Zhang, J. Liu, X. Zhang, and Y. L. Lu. Multiscale channel attention-driven graph dynamic fusion learning method for robust fault diagnosis. *IEEE Transactions on Industrial Informatics*, 20(9):11002–11013, 2024.

[23] B. Zhao, X. M. Zhang, Z. H. Zhan, Q. Q. Wu, and H. D. Zhang. Multiscale graph-guided convolutional network with node attention for intelligent health state diagnosis of a 3-PRR planar parallel manipulator. *IEEE Transactions on Industrial Electronics*, 69(11):11733–11743, 2021.

[24] Y. D. Xu, S. Li, Z. H. Li, K. L. Wu, R. Y. Huang, B. B. Sun, and J. Ji. Time–frequency fully-connected graph neural network: An effective multiscale spatiotemporal dependency learning method for multisource machine fault diagnosis. *Advanced Engineering Informatics*, 69:104023, 2026.

[25] X. B. Li, B. Xiao, X. J. Chen, P. Jiang, and et al. MSDF-VAE: A cloud–edge collaborative method for fault diagnosis based on transfer learning. *IEEE Internet of Things Journal*, 12(12):22393–22403, 2025.

[26] S. H. Liu, J. H. Zhang, W. D. Huang, F. Lyu, D. D. Wang, and B. Xu. Temporal-spatial attention network: A novel axial piston pump coupled fault diagnosis method. *IEEE Transactions on Instrumentation and Measurement*, 73:1–15, 2024.

[27] S. Z. Li, W. C. Meng, S. B. He, J. C. Bi, and G. L. Liu. Staged: A spatial-temporal aware graph encoder–decoder for fault diagnosis in industrial processes. *IEEE Transactions on Industrial Informatics*, 20(2):1742–1752, 2023.

[28] C. Y. Yang, K. B. Zhou, and J. Liu. Supergraph: Spatial-temporal graph-based feature extraction for rotating machinery diagnosis. *IEEE Transactions on Industrial Electronics*, 69(4):4167–4176, 2021.

[29] V. Sadhu, K. Anjum, and D. Pompili. On-board deep-learning-based unmanned aerial vehicle fault cause detection and classification via FPGAs. *IEEE Transactions on Robotics*, 39(4):3319–3331, 2023.

[30] K. He, D. J. Yu, D. Wang, M. J. Chai, S. T. Lei, and C. L. Zhou. Graph attention network-based fault detection for UAVs with multivariant time series flight data. *IEEE Transactions on Instrumentation and Measurement*, 71:1–13, 2022.

[31] L. Tang, A. Wang, B. S. Xia, Y. C. Tang, and Q. B. Chen. Research on integrated sensing, communication resource allocation, and digital twin placement based on digital twin in IoV. *IEEE Internet of Things Journal*, 12(11):17300–17315, 2025.

[32] J. Y. Xia, R. Y. Huang, J. P. Li, Z. Y. Chen, and W. H. Li. Digital twin-assisted fault diagnosis of rotating machinery without measured fault data. *IEEE Transactions on Instrumentation and Measurement*, 73:1–10, 2024.

[33] S. Li, Q. B. Jiang, Y. D. Xu, K. Feng, Y. L. Wang, B. B. Sun, X. A. Yan, X. Sheng, K. Zhang, and Q. Ni. Digital twin-driven focal modulation-based convolutional network for intelligent fault diagnosis. *Reliability Engineering & System Safety*, 240:109590, 2023.

[34] W. Y. Hu, T. Y. Wang, and F. L. Chu. Novel Ramanujan digital twin for motor periodic fault monitoring and detection. *IEEE Transactions on Industrial Informatics*, 19(12):11564–11572, 2023.

[35] C. C. Peng and Y. H. Chen. Fixed-wing unmanned aerial vehicle rotary engine anomaly detection via online digital twin methods. *IEEE Transactions on Aerospace and Electronic Systems*, 60(1):741–758, 2024.

[36] Z. Ming, B. P. Tang, L. Deng, Q. C. Yang, and Q. K. Li. Digital twin-assisted fault diagnosis framework for rolling bearings under imbalanced data. *Applied Soft Computing*, 168:112528, 2025.

[37] Q. B. Wang, X. H. Zhang, J. W. Lu, G. Xiao, Y. X. Ren, and W. J. Li. Digital twin-driven physically constrained generative adversarial network for industrial boiler fault diagnosis. *IEEE Transactions on Instrumentation and Measurement*, 74:1–15, 2025.

[38] H. Li, G. L. Yang, Q. Z. Cai, and Y. Q. Tu. Digital-twin-assisted fault diagnosis for IMU in rotational inertial navigation systems. *IEEE Sensors Journal*, 26(1):849–859, 2025.

[39] J. Song, D. W. Wang, Z. P. Chen, and K. Zhao. Digital twin system for vtol UAV fault diagnosis based on px4. In *International Conference on Autonomous Unmanned Systems*, pages 2389–2401. Springer, 2022.

[40] J. H. Tu, X. H. Nian, and X. H. Dai. Physics-aware multi-channel vector quantization for hybrid digital twin modeling of UAV systems. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–18, 2025. Doi:10.1109/TAES.2025.3639005.

[41] X. L. Le, B. Jin, G. Cui, X. H. Dai, and Q. Quan. RflyMAD: A dataset for multicopter fault detection and health assessment. *The International Journal of Robotics Research*, 44(7):1081–1092, 2025.

Authorized licensed use limited to: Central South University. Downloaded on July 27,2026 at 05:00:59 UTC from IEEE Xplore. Restrictions apply.
© 2026 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.