Physics-Aware Multichannel Vector Quantization for Hybrid Digital Twin Modeling of UAV Systems

JINHU TU
XIAOHONG NIAN
Central South University, Changsha, China

XUNHUA DAI
Central South University, Changsha, China

As Big Data technologies continue to transform industrial manufacturing, digital twin (DT) systems powered by artificial intelligence are becoming key enablers in shaping the emerging low-altitude economy. Among these, unmanned aerial vehicle (UAV) systems play a central role. However, building reliable DTs for UAVs remains challenging due to the heterogeneous nature of sensor data, complex spatiotemporal distributions, and nonlinear interactions within system dynamics. In this work, we propose a fusion framework for UAV-oriented DT modeling that integrates physical priors with data-driven learning, tailored for enhanced DT modeling of UAVs. The method is centered on a multichannel soft vector quantization mechanism, which performs feature clustering across heterogeneous sensor modalities and time scales, enabling robust fusion of multisource data. By embedding physics-aware constraints and leveraging a hybrid fusion strategy, the model enhances generalization under real-world uncertainties. We validate our method by developing a complete UAV DT framework and collecting a large-scale dataset that integrates both simulated and real flight data. Extensive evaluations across hovering, takeoff, tour, and forward flight missions demonstrate superior fidelity and robustness compared to conventional DT models. This work contributes a scalable UAV DT architecture with strong fusion capability and provides open-source tools for advancing self-evolving DT systems in dynamic environments.

Received 21 July 2025; revised 25 October 2025; accepted 26 November 2025. Date of publication 1 December 2025; date of current version 9 February 2026.

DOI. No. 10.1109/TAES.2025.3639005

Refereeing of this contribution was handled by Q. Zhu.

This work was supported in part by the National Natural Science Foundation of China under Grant 62406345, and in part by the Natural Science Foundation of Hunan Province, China under Grant 2025JJ50341.

Authors' addresses: Jinhu Tu and Xiaohong Nian are with the School of Automation, Central South University, Changsha 410083, China, E-mail: (tjhcsu@csu.edu.cn; xhnian@csu.edu.cn); Xunhua Dai is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China, E-mail: (dai.xh@csu.edu.cn). *(Corresponding author: Xunhua Dai.)*

I. INTRODUCTION

A. Background

The convergence of Industry 4.0, artificial intelligence (AI), and next-generation communication technologies is propelling the evolution of low-altitude operational ecosystems [1], [2], in which unmanned aerial vehicles (UAVs) play a central role in mission-critical applications such as intelligent inspection, aerial logistics, and infrastructure surveillance [3], [4], [5]. As integral nodes within the emerging Artificial Intelligence of Things paradigm, UAVs are increasingly expected to operate autonomously in uncertain and dynamic environments, requiring robust perception, situational awareness, and adaptive decision-making capabilities [6]. Digital twin (DT) technology [7], [8] has emerged as a key enabler of such capabilities by establishing high-fidelity virtual counterparts that facilitate real-time state estimation, predictive diagnostics, and coordinated cyber-physical control [9], [10]. While promising, conventional DT modeling approaches often rely on static or oversimplified physical priors, thereby limiting their capacity to accurately characterize the nonlinear, multimodal, and multiscale behaviors inherent in UAV sensor networks and flight dynamics [11], [12]. This modeling gap poses significant challenges for system resilience and reliability, particularly in the context of safety-critical aerospace operations. Addressing these limitations necessitates hybrid DT modeling frameworks that integrate domain-informed priors with data-driven learning and multisource sensor fusion. Such approaches can significantly enhance the representational fidelity, generalization ability, and adaptive responsiveness of UAV-based DT systems, thereby advancing autonomous operations in complex low-altitude airspace environments.

B. Related Work and Research Motivation

DT technology establishes a real-time bidirectional mapping between the physical and digital domains through the construction of high-fidelity virtual counterparts, offering a novel paradigm for UAV system modeling [13]. These models integrate multiple representational dimensions—geometric, physical, behavioral, and operational—to capture system–environment interactions across scales [14]. Among these components, simulation modeling constitutes the core of DT, as it plays a critical role in capturing complex UAV dynamics, structural responses, and environmental perturbations during operation.

In the UAV domain, DT development remains in an exploratory phase [12], [13], [14]. Existing studies have primarily focused on the realization of DT platforms and system-level frameworks, including mission planning, flight visualization, swarm collaboration, and cyber–physical interaction [30], [31]. Such research provides valuable experimental environments and interface support for UAV mission execution. However, precise modeling of complex dynamics and heterogeneous multisensor data remains relatively underdeveloped—particularly in terms of cross-modal feature fusion and high-dimensional

data-driven modeling. Current UAV-DT modeling approaches can be broadly categorized into the following two paradigms.

1) *Physics-based modeling:* These methods rely on multibody dynamics, aerodynamics, and control equations to explicitly describe UAV behavior [15], [16]. They exhibit strong interpretability and physical generalizability but often make simplified assumptions regarding complex nonlinear and multiscale interactions. As a result, they tend to have high computational costs and face difficulties in parameter identification, limiting their adaptability in highly uncertain or disturbed environments [20], [21].
2) *Data-driven modeling:* Leveraging deep neural networks, these approaches can automatically extract high-dimensional and temporal features from flight logs and sensor data [17], [18], [19], [22], [23]. They demonstrate greater flexibility and self-adaptation in complex environments, yet typically depend on large volumes of high-quality data. The lack of physical constraints and interpretability makes them prone to overfitting and instability, especially under sparse data or distribution-shift conditions.

Because neither paradigm alone can fully reconcile mechanistic accuracy, real-time performance, and generalization capability, hybrid modeling approaches have emerged. These methods integrate physical mechanisms with data-driven learning to combine interpretability and adaptability. For example, Yin et al. [24] and Kang et al. [25] proposed hybrid DT architectures that fuse physical modeling with machine learning for adaptive control and prediction; Waseem et al. [26] developed a DT surrogate framework for multiscale state monitoring in manufacturing processes; and Liu et al. [27] introduced a modular DT building strategy that enables standardized and scalable twin system construction through staged modeling. Despite these theoretical and architectural advances, real-world UAV applications often involve multilevel, highly coupled interactions—including sensor fusion, task scheduling, and dynamic control—that exhibit strong nonlinearity and time variance under extreme environments [28], [29]. Consequently, existing DT models still struggle to maintain dynamic consistency and high-confidence mapping between physical and digital spaces, limiting their applicability in dynamic and uncertain scenarios.

Among the aforementioned challenges, data scarcity and distribution imbalance remain key bottlenecks constraining the fidelity and generalization of UAV DTs. The generation of reliable synthetic data offers an effective way to mitigate the insufficiency of real-world data, thereby enhancing model robustness and adaptability [30], [31]. However, conventional methods often fail under sparse operational conditions, extreme environments, or unseen mission scenarios. Generative modeling provides a promising pathway to bridge this gap [32], [33] by producing and augmenting high-quality simulated data that support data-driven DT modeling. Deep generative models—such as variational autoencoders (VAEs) [34] and generative adversarial networks (GANs) [35]—have been widely employed for data augmentation and feature learning. These methods effectively address data scarcity and improve model generalization [36], [37], [38]. Nonetheless, their applications in UAV DTs face the following two major limitations.

1) *Single-modality and localized modeling:* Most existing approaches focus on modeling data from a single sensor under fixed conditions, optimizing for specific physical parameters [39]. In contrast, UAVs are complex, tightly coupled systems integrating propulsion, control, sensing, and communication subsystems, characterized by multisource heterogeneity and multiscale data structures—imposing higher demands on generative modeling.
2) *Multiscale distribution inconsistency:* Significant discrepancies exist among data distributions from different sensors and task phases, often leading to mode collapse or overfitting during generative model training, thereby restricting generalization performance [40].

Therefore, there is an urgent need for a unified generative framework capable of integrating multimodal and multiscale information, while maintaining stable generative performance under sparse and uncertain conditions. This article aims to address this problem by proposing a generative hybrid modeling framework that combines physical constraints with data-driven generative mechanisms, targeting high-fidelity, adaptive UAV DT modeling under data-scarce and uncertain operational environments.

C. Main Contributions

In this work, we propose a DT-enhanced modeling framework for UAVs that integrates physical knowledge through simulation-based data generation and error-driven supervision, thereby maintaining physics-aware modeling logic. As illustrated in Fig. 1(a), the framework consists of real UAV nodes [see Fig. 1(i)] and their corresponding DT nodes [see Fig. 1(iii)]. For the virtual nodes, a hierarchical DT model is constructed based on physics-aware principles, augmented by a rendering engine and multiresolution modeling strategy. This enables high-fidelity mapping from system-level abstractions to subsystem-level dynamics. A parallel-twin mechanism [see Fig. 1(ii)] is implemented onboard the real UAVs, allowing for real-time synchronization and bidirectional data exchange between the physical and virtual domains. To address the challenges posed by heterogeneous, multiscale UAV data, we further introduce a multichannel soft vector-quantized variational autoencoder (VQVAE), as shown in Fig. 1(b). This module leverages discrete feature encoding and soft quantization to capture nonlinear, scale-varying error patterns, thereby enhancing the fusion and abstraction of multimodal sensor data. The model is trained end-to-end, maintaining generative fidelity and generalization even under sparse, uncertain, or previously unseen conditions. The framework is validated using

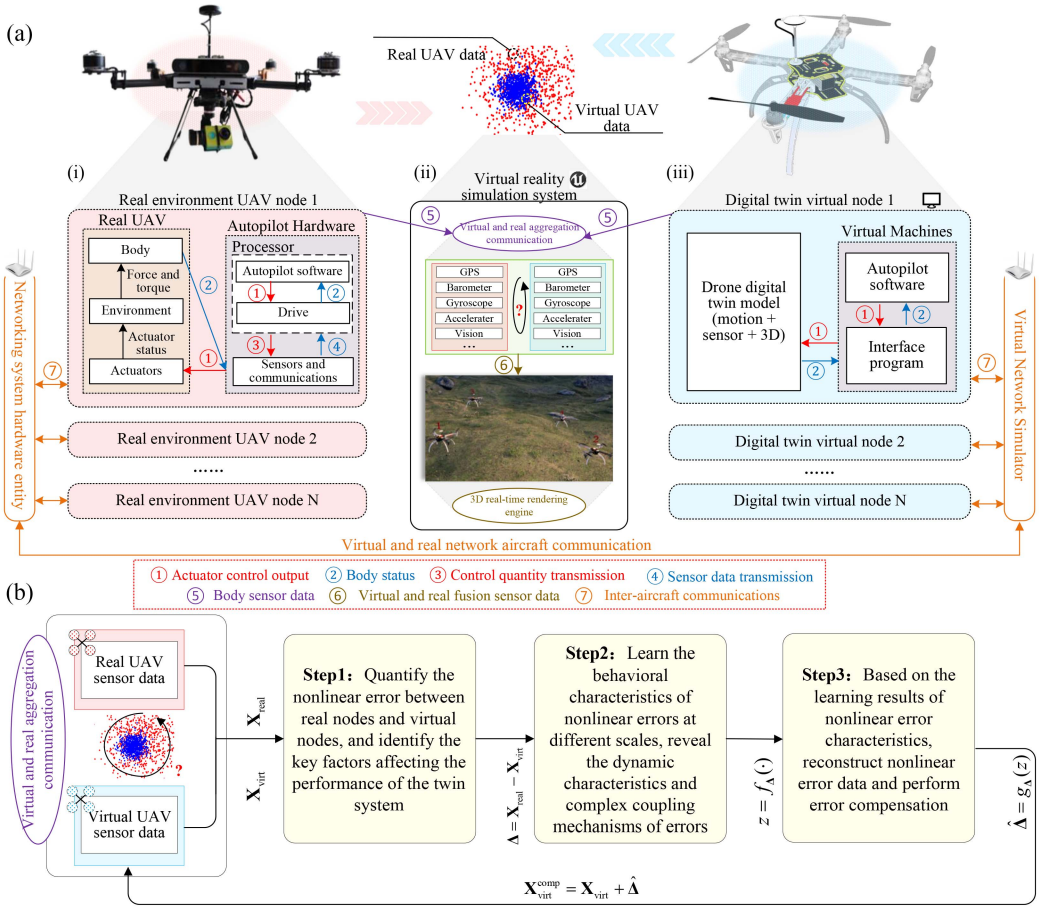


Fig. 1. UAV DT application framework and DT-enhanced modeling method. (a) DT-based UAV virtual–real interactive communication framework: (i) Composition of the UAV real node system and its underlying communication framework, (ii) virtual–real synchronous communication framework of UAVs and their underlying data interaction, and (iii) composition of the UAV virtual node system and its underlying simulation-based communication framework. (b) DT-enhanced modeling process, where $\mathbf{X}_{\text{real}}$ is the real node data, $\mathbf{X}_{\text{virt}}$ is the virtual node data, $\mathbf{\Delta}$ is the error between the real and virtual, $f_{\mathbf{\Delta}}(\cdot)$ is the function that describes the error behavior, z represents the characteristics of the error in the latent space, and $g_{\mathbf{\Delta}}(\cdot)$ is the function used for error reconstruction.

real-world UAV flight data across diverse missions, flight modes, and environmental conditions. Experimental results demonstrate significant improvements in reconstruction accuracy, data consistency, and cross-modal representation compared to baseline methods, highlighting the framework's potential for lifecycle modeling, anomaly characterization, and adaptive control in complex operational scenarios. The main contributions of this article are as follows.

1) A DT-based UAV modeling framework is proposed, which integrates physics-aware system representations with end-to-end generative learning. This framework implements a parallel twin mechanism, enabling real-time bidirectional data interaction between the physical UAV and its DT. As a result, system modeling accuracy and representational capacity are enhanced under complex dynamic conditions, while high-fidelity synchronization of virtual and real data are supported.
2) A multichannel soft VQVAE is proposed to efficiently process multisource, multimodal, and heterogeneous UAV sensor data. Leveraging independent channel embeddings, flight-state-conditioned encoding, and soft quantization mechanisms, the autoencoder achieves efficient decoupling and encoding of multimodal features, captures nonlinear error patterns, and enables high-precision data reconstruction with cross-modal consistency.
3) An open-source DT platform, along with a dedicated multimodal UAV dataset, is made publicly available. This resource not only guarantees the reproducibility of the proposed methods but also provides standardized tools and datasets for the testing, validation, and further advancement of intelligent UAV modeling techniques.

The rest of this article is organized as follows. Section II presents the proposed methodology, including problem analysis, nonlinear error modeling, and the DT modeling process based on feature clustering and multichannel soft vector quantization. Section III presents the experimental results, covering dataset details, evaluation setup, and

comprehensive comparisons, including real-flight verification and ablation studies. Finally, Section IV concludes this article.

II. METHODS

A. Process Characteristics and Problem Analysis

Fig. 1(a) illustrates the application framework of UAV DTs, where virtual nodes replicate UAV hardware behavior by running autopilot software and physical models within a virtual environment. In contrast, real nodes rely on onboard autopilot hardware and airframes to execute flight tasks. Real-time data exchange between virtual and real nodes via communication protocols ensures synchronization and supports decision making during mission execution.

In this process, data from the virtual nodes are crucial for making decisions in upstream tasks, with their reliability—relative to the real nodes—playing a vital role in task success. However, these data typically present the following characteristics.

1) *Multimodal:* UAV missions vary widely, leading to multimodal data across different flight modes such as takeoff, hover, and forward. Each mode exhibits distinct dynamic characteristics and sensor responses, requiring careful handling of these differences in modeling.
2) *Multiscale:* UAVs use multiple sensors [e.g., global positioning system (GPS) and inertial measurement unit (IMUs)], each generating data across varying temporal, spatial, and magnitude scales. The error models and data distributions from different sensors can differ significantly, further complicating data integration.
3) *Nonlinear coupling:* UAV system behavior is affected by a variety of factors, such as aerodynamic effects, sensor errors, and external conditions (e.g., weather). These factors interact in complex nonlinear ways, which cannot be fully captured by traditional simplified physical models.

Given these challenges, particularly under varying flight modes and extreme environmental conditions, it is crucial to manage multimodal and multiscale data effectively in order to enhance DT-based modeling. Simplified physical models often fail to capture the nonlinear couplings between environmental changes and system performance, highlighting the need for advanced modeling techniques. Integrating physical priors with learning-based methods offers a promising path to improve prediction fidelity and adaptive control in UAV systems.

B. Scheme Design and Objective of Solution

As shown in Fig. 1(b), the key steps in the enhanced DT modeling of UAVs are presented. The core idea is to improve the accuracy and reliability of the DT model by precisely capturing and compensating for the nonlinear coupling between the real environment and system performance. The specific design of the solution involves the following three key steps.

1) *Quantifying the Nonlinear Errors:* Let $\mathbf{X}_{\text{real}} \in \mathbb{R}^{d_{\text{real}}}$ represent the state data of the real node, and $\mathbf{X}_{\text{virt}} \in \mathbb{R}^{d_{\text{virt}}}$ represent the state data of the virtual node, where d_{real} and d_{virt} denote the dimensions of the real and virtual data, respectively. The nonlinear error can be expressed as

$$\mathbf{\Delta} = \mathbf{X}_{\text{real}} - \mathbf{X}_{\text{virt}} \tag{1}$$

where $\mathbf{\Delta}$ represents the error between the real and virtual nodes. These errors arise not only from sensor accuracy, noise, and data latency, but also from environmental factors affecting the UAV system's performance, such as meteorological changes, flight modes, and external disturbances.

2) *Learning the Behavior of Nonlinear Errors:* A data-driven learning framework is introduced to capture the nonlinear structure of the errors. Let $f_{\mathbf{\Delta}}(\cdot)$ be the function that describes the error's behavior at different scales. The error $\mathbf{\Delta}$ is mapped to a lower dimensional latent space as follows:

$$z = f_{\mathbf{\Delta}}(\Delta|\theta) \tag{2}$$

where $z \in \mathbb{R}^{d_z}$ represents the feature of the error in the latent space, and $f_{\mathbf{\Delta}}(\cdot)$ is a nonlinear mapping function. The goal is to reveal the dynamic variation of the errors under different flight phases and environmental conditions. By adopting deep learning and other methods, the model learns the transmission and mutual influence mechanisms of the errors across various levels of the system.

3) *Reconstruction and Compensation of Nonlinear Errors:* The latent features z learned in step 2 are mapped back to the original error space through a generative model, thereby reconstructing the error between the real and virtual nodes

$$\tilde{\mathbf{\Delta}} = g_{\mathbf{\Delta}}(z) \tag{3}$$

where $g_{\mathbf{\Delta}}(\cdot)$ is the decoding function for error reconstruction, and $\tilde{\Delta}$ represents the reconstructed nonlinear error. The reconstructed error is then used to compensate the virtual node data

$$\mathbf{X}_{\text{virt}}^{\text{comp}} = \mathbf{X}_{\text{virt}} + \tilde{\Delta} \tag{4}$$

where $\mathbf{X}_{\text{virt}}^{\text{comp}}$ is the enhanced DT model, which incorporates the nonlinear error information of the system. This enhanced model more accurately reflects the real system's performance, improving its effectiveness in complex application scenarios.

C. DT Modeling and Network Design

In response to the objectives outlined in Section II-B, this section presents a data reconstruction method designed to capture the nonlinear coupling between the real environment and system performance and to effectively compensate for errors in the DT model. As shown in Fig. 2, the proposed method integrates DT technology, feature clustering, and vector quantization to enhance UAV modeling and data generation. In the DT modeling stage [see Fig. 2(a)], a base UAV model is developed, which contains key components

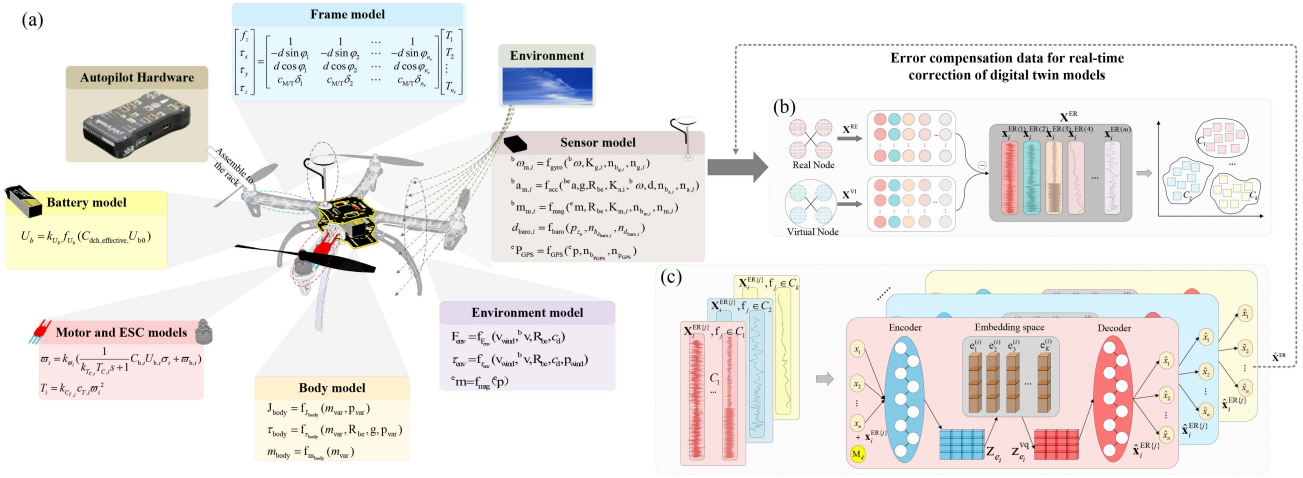


Fig. 2. Framework of the UAV DT-enhanced multiscale sensor modeling method. (a) UAV DT modeling framework (for modeling details of different submodules, visit https://github.com/RflySim/RflySimModel). (b) Illustration of dataset construction and feature clustering based on the DT model and real-world data. (c) Multichannel multiscale data modeling network. Here, $\mathbf{X}^{\mathrm{RE}}$ represents the flight dataset of real nodes, $\mathbf{X}^{\mathrm{VI}}$ represents the flight dataset of virtual nodes, $\mathbf{X}^{\mathrm{ER}}$ represents the error dataset, $\mathbf{f}_j$ is the feature vector, C_l represents the lth cluster, M_d is the modal information, e is the discrete embedding space, and z_{e_i} is the latent representation.

such as dynamics, control systems, battery management, sensors, structural elements, and environmental perturbations. The model can accurately simulate the behavior of UAVs under complex conditions and provide physical prior data input for subsequent learning-based methods. In the feature clustering and dataset construction stage [see Fig. 2(b)], real and virtual sensor data are combined. Feature clustering methods improve data generation accuracy by capturing variations across different modalities and scales. In the multichannel, multiscale data modeling stage [see Fig. 2(c)], vector quantization discretizes high-dimensional sensor data. Integrating feature clustering results enables multiscale data fusion and compensation, improving the precision and generalizability of the DT model.

1) *DT-Level Modeling:* The rigid-body model of UAV flight control, as illustrated in Fig. 2(a), can be finally formulated as follows:

$$
\begin{aligned}
&{}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} \\
&{}^{\mathrm{e}}\mathbf{v} = \mathbf{R}_{\mathrm{be}}^{\mathrm{T}}\,{}^{\mathrm{b}}\mathbf{v} \\
&{}^{\mathrm{b}}\dot{\mathbf{v}} = -\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\cdot{}^{\mathrm{b}}\mathbf{v} + g\cdot\mathbf{R}_{\mathrm{be}}^{\mathrm{T}}\mathbf{e}_3 + \frac{\mathbf{F}}{m} \\
&\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -{}^{\mathrm{b}}\boldsymbol{\omega}^{\mathrm{T}} \\ {}^{\mathrm{b}}\boldsymbol{\omega} & -[{}^{\mathrm{b}}\boldsymbol{\omega}]_{\times} \end{bmatrix} q \\
&\mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega}\times(\mathbf{J}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}) + \boldsymbol{\tau} + \mathbf{G}_a
\end{aligned}
\tag{5}
$$

where ${}^{\mathrm{e}}\mathbf{p}$, ${}^{\mathrm{e}}\mathbf{v} \in \mathbb{R}^3$ represent the UAV's position and velocity in the world frame, and ${}^{\mathrm{b}}\mathbf{v}$ denotes the velocity in the body frame. The matrix $\mathbf{R}_{\mathrm{be}}^{\mathrm{T}}$ is the rotation matrix, while m represents the mass of the multirotor UAV. The term $\mathbf{e}_3$ is the unit vector along the z-axis in the inertial frame, and $\mathbf{F}$ denotes the total thrust generated by all rotors. The variable ${}^{\mathrm{b}}\boldsymbol{\omega}$ represents the body angular velocity, and g is the gravitational acceleration. The term $\boldsymbol{\tau} = [\tau_x \tau_y \tau_z]^{\mathrm{T}} \in \mathbb{R}^3$ corresponds to the torque generated by the rotor thrust along the body axes. The matrix $\mathbf{J} \in \mathbb{R}^{3\times3}$ represents the moment of inertia of the UAV, and $\mathbf{G}_a = [\mathbf{G}_{a,\phi}\mathbf{G}_{a,\theta}\mathbf{G}_{a,\psi}]^{\mathrm{T}} \in \mathbb{R}^3$ accounts for the gyroscopic torque.

REMARK 1 While the rigid-body dynamics model provides a solid foundation for describing UAV flight behavior, real-world applications introduce uncertainties such as environmental disturbances, sensor noise, and modeling errors. These factors collectively contribute to deviations from idealized predictions. A true DT must accurately represent the UAV throughout its entire lifecycle. However, relying solely on a rigid-body model is insufficient for high-precision DT applications. To minimize discrepancies between theoretical models and actual flight performance, it is crucial to incorporate error modeling and correction mechanisms, ensuring a more accurate and reliable representation of UAV behavior.

2) *Feature Clustering:* As shown in Fig. 2(b), let $\mathbf{X}^{\mathrm{RE}} = \{\mathbf{x}_1^{\mathrm{RE}}, \mathbf{x}_2^{\mathrm{RE}}, \ldots, \mathbf{x}_n^{\mathrm{RE}}\}$ represent the flight dataset of the real node, and $\mathbf{X}^{\mathrm{VI}} = \{\mathbf{x}_1^{\mathrm{VI}}, \mathbf{x}_2^{\mathrm{VI}}, \ldots, \mathbf{x}_n^{\mathrm{VI}}\}$ represent the flight dataset of the virtual node, where n denotes the number of data samples. Each sample, $\mathbf{x}_i^{\mathrm{RE}}$, consists of m data dimensions, $\mathbf{x}_i^{\mathrm{RE}} = \{x_i^{\mathrm{RE}(1)}, x_i^{\mathrm{RE}(2)}, \ldots, x_i^{\mathrm{RE}(m)}\}$. The real node data are obtained from actual UAV sensor measurements, while the virtual node data are derived from the simulation results of the DT UAV. The resulting error dataset can be represented as $\mathbf{X}^{\mathrm{ER}} = \{\mathbf{x}_1^{\mathrm{ER}}, \mathbf{x}_2^{\mathrm{ER}}, \ldots, \mathbf{x}_n^{\mathrm{ER}}\}$, where each error sample $\mathbf{x}_i^{\mathrm{ER}}$ is the difference between the corresponding real and virtual node data: $\mathbf{x}_i^{\mathrm{ER}} = \mathbf{x}_i^{\mathrm{RE}} - \mathbf{x}_i^{\mathrm{VI}}$.

To capture the distributional differences across multiple dimensions of the data, we first perform feature clustering on each dimension of the dataset $\mathbf{X}^{\mathrm{RE}}$. For the jth dimension,

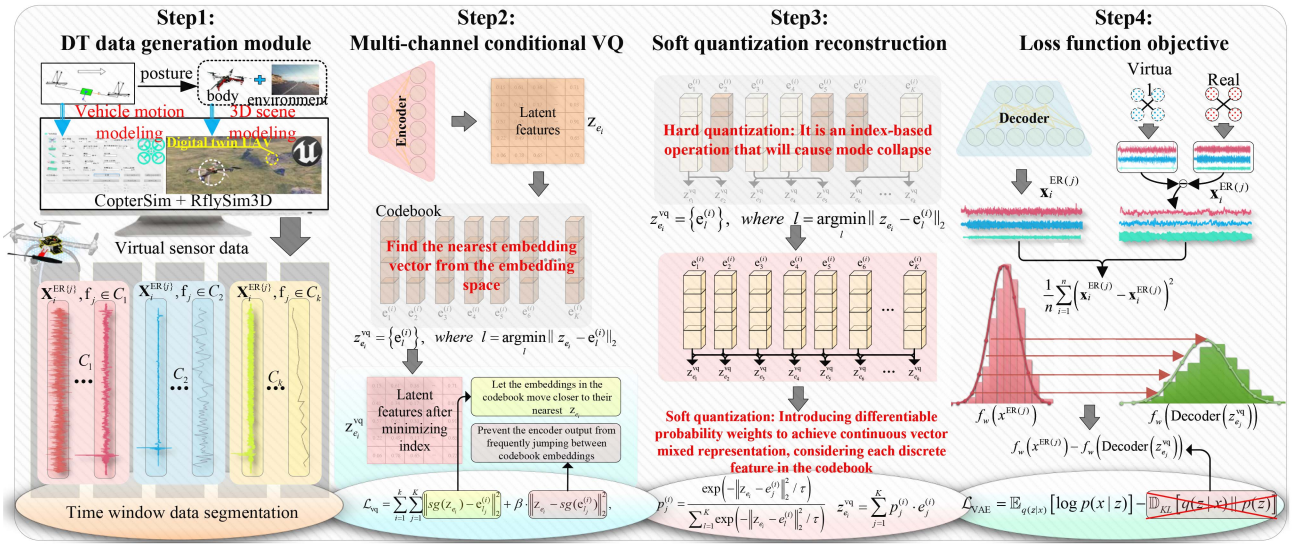


Fig. 3. Illustration of the enhanced modeling framework based on DTs. Step 1 corresponds to Section II-C1, Step 2 corresponds to Section II-C3, Step 3 corresponds to Section II-C4, and Step 4 is the overall loss function objective. Here, $\mathbf{f}_j$ is the feature vector, C_l represents the lth cluster, M_d is the modal information, e is the discrete embedding space, z_{e_i} is the potential representation, $z_{e_i}^{\text{vq}}$ is the quantized representation, sg$(\cdot)$ denotes the stop-gradient operation, and $p_j^{(i)}$ indicates the relevance between z_{e_i} and the discrete vector $e_j^{(i)}$.

we define the feature vector $\mathbf{f}_j$ as follows:

$$\mathbf{f}_j=\left[\mu_j,\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i^{\text{ER}(j)}-\mu_j\right)^2,\frac{\sum_{i=1}^{n-\ell}\left(\mathbf{x}_i^{\text{ER}(j)}-\mu_j\right)\left(\mathbf{x}_{i+\ell}^{\text{ER}(j)}-\mu_j\right)}{\sum_{i=1}^{n}\left(\mathbf{x}_i^{\text{ER}(j)}-\mu_j\right)^2}\right] \tag{6}$$

where $j=1,2,\ldots,m$ and μ_j is the mean of the jth dimension: $\mu_j=\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^{\text{ER}(j)}$. The second term represents the variance, and the third term is the autocorrelation coefficient, where ℓ is the time lag index of the autocorrelation term. The feature vector $\mathbf{f}_j$ is then used as input for clustering. The objective is to minimize the total squared error within each cluster using k-means clustering, as defined by the following cost function:

$$J=\sum_{l=1}^{k}\sum_{\mathbf{f}_j\in C_l}\|\mathbf{f}_j-c_l\|^2 \tag{7}$$

where C_l denotes the lth cluster, and c_l is the center of the lth cluster. Each feature vector $\mathbf{f}_j$ is assigned to the cluster that minimizes the distance to its center:

$$\mathbf{f}_j\in C_l,\ \text{where } l=\underset{l}{\text{argmin}}\|\mathbf{f}_j-c_l\|. \tag{8}$$

As shown in Fig. 2(b), the final dataset can be represented as $\mathbf{X}^{\text{ER}}=\{(\mathbf{x}_i^{\text{ER}(j)},C_l)\}_{i=1,\ldots,n}^{l=1,\ldots,k}$, where each data point $\mathbf{x}_i^{\text{ER}(j)}$ is assigned to its corresponding cluster C_l based on the clustering results.

3) *Multichannel Conditional VQ:* In traditional vector quantization (VQ) methods, input data are mapped to a latent space $z_e(x)$ by an encoder, and a discrete embedding space $e\in\mathbb{R}^{K\times D}$ (also known as the codebook) is used to discretize the latent representation. The core step in this process involves replacing the original latent representation with the closest discrete vector found in the embedding space. In multichannel VQ, we incorporate feature clustering results, where k clusters C_i are obtained through feature clustering. As illustrated in Fig. 2(c), for each cluster C_i, an independent embedding space $e_i=\{\mathrm{e}_1^{(i)},\mathrm{e}_2^{(i)},\ldots,\mathrm{e}_K^{(i)}\}$ is used. This embedding space contains K discrete vectors, where $\mathrm{e}_l^{(i)}\in\mathbb{R}^D$, for $l=1,\ldots,K$, and the latent representation z_{e_i}, where $i=1,\ldots,k$, is quantized to its corresponding embedding space e_i. Furthermore, as shown in *Step 2* of Fig. 3, the quantization operation maps the encoder output z_{e_i} to the nearest discrete embedding vector $\mathrm{e}_l^{(i)}$, resulting in the quantized representation

$$z_{e_i}^{\text{vq}}=\left\{\mathrm{e}_l^{(i)}\right\},\ \text{where } l=\underset{l}{\text{argmin}}\|z_{e_i}-\mathrm{e}_l^{(i)}\|_2 \tag{9}$$

where l represents the index of the closest discrete vector in the embedding space e_i to z_{e_i}. To optimize the embedding space, the objective function for VQ can be expressed as

$$\mathcal{L}_{\text{vq}}=\sum_{i=1}^{k}\sum_{j=1}^{K}\left\|\text{sg}(\mathrm{z}_{e_i})-\mathrm{e}_{l_j}^{(i)}\right\|_2^2+\beta\cdot\left\|\mathrm{z}_{e_i}-\text{sg}(\mathrm{e}_{l_j}^{(i)})\right\|_2^2 \tag{10}$$

where sg$(\cdot)$ denotes the stop-gradient operation. The first term represents the codebook loss, which measures the difference between the quantized embedding vector $e_l^{(i)}$ and the encoder's output. The second term is the commitment loss, ensuring that the encoder output z_{e_i} remains close to the discrete embedding vectors $e_l^{(i)}$ during training. The hyperparameter β controls the balance between the two loss terms. To account for the effects of different flight modes on the sensor data, we introduce modal information M_d. Thus, the final data representation is expressed as $\mathbf{X}^{\text{ER}}=\{(\mathbf{x}_i^{\text{ER}\{j\}},C_l,M_d)\}$, where $d=1,\ldots,p$, and p represents the number of UAV flight modes. Under this condition, the

Algorithm 1: DT-Enhanced Modeling Based on Feature Clustering and Multichannel Soft Vector Quantization.

Input:
- Input data tensor $X \in \mathbb{R}^{B\times T\times N}$ (batch size B, sequence length T, feature dimension N)
- Conditional variable $C \in \mathbb{R}^{B\times 1}$

Output:
- Reconstructed data $\hat{X}$
- Loss components $\mathcal{L}_{\text{total}}$, $\mathcal{L}_{\text{recon}}$, $\mathcal{L}_{\text{vq}}$, $\mathcal{L}_W$

procedure ForwardPass$X$, C
  $C_{\text{emb}} \leftarrow \text{Repeat}(f_{\text{cond}}(C), [1, T, 1])$ ▷ Expand condition to sequence length
  for $i \leftarrow 1$ to M **do** ▷Process each feature split
    $X_i \leftarrow X[:, :, S_i]$
    $Z_i \leftarrow \text{LSTM}^i_{\text{enc}}([X_i \| C_{\text{emb}}])$ ▷$\|$ denotes concatenation
  end for
  for $i \leftarrow 1$ to M **do**
    $D_i \leftarrow \text{CDist}(Z_i^{\text{flat}}, E_i)^2$ ▷Pairwise squared distances
    $A_i \leftarrow \text{Softmax}(-D_i)$ ▷Soft assignments
    $\hat{Z}_i \leftarrow \text{Reshape}(A_i E_i^T, [B, T, D])$
    $\mathcal{L}^i_{\text{vq}} \leftarrow \|Z_i - \hat{Z}_i\|_2^2 + \beta\|\text{sg}(Z_i) - E_i\|_2^2$
    $\hat{Z}_i^{\text{out}} \leftarrow Z_i + (\hat{Z}_i - Z_i).\text{detach}()$ ▷Straight through estimator
  end for
  for $i \leftarrow 1$ to M **do**
    $\hat{X}_i \leftarrow \text{LSTM}^i_{\text{dec}}([\hat{Z}_i^{\text{out}} \| C_{\text{emb}}])$
    $\mathcal{L}^i_{\text{recon}} \leftarrow \|X_i - \hat{X}_i\|_2^2$
  end for
  $\mathcal{L}_W \leftarrow \|\mu_X - \mu_{\hat{X}}\|_1 + \|\sigma_X - \sigma_{\hat{X}}\|_1$
  $\mathcal{L}_{\text{total}} \leftarrow \sum_i \mathcal{L}^i_{\text{recon}} + \sum_i \mathcal{L}^i_{\text{vq}} + \mathcal{L}_W$
end procedure
procedure Training(Dataset $\mathcal{D}$, Epochs E)
  while not converged **do**
    $(X, C) \leftarrow \text{SampleBatch}(\mathcal{D})$
    $\hat{X}, \mathcal{L}_{\text{total}} \leftarrow \text{ForwardPass}(X, C)$
    $\theta \leftarrow \theta - \alpha \cdot \text{AdamW}(\nabla_\theta \mathcal{L}_{\text{total}})$
    $\|\theta\| \leftarrow \text{Clip}(\|\theta\|, -c, c)$
  end while
end procedure

embedding space e_i is adjusted and optimized based on the modal information M_d, thereby generating discrete features specific to each flight mode.

4) *Soft Quantization Reconstruction:* In the VQ quantization process, as shown in (9), discrete quantization is achieved by minimizing the distance between the continuous latent representation and the discrete embedding vector, a process known as hard quantization. However, during training, some discrete vectors may be selected frequently while others are rarely chosen. This imbalance can lead to certain vectors in the embedding space being inadequately updated, resulting in mode collapse. To address this, we introduce a differentiable probability distribution to replace the discrete selection in hard quantization. The underlying mechanism is shown in *Step 3* of Fig. 3. Specifically, the distance between the latent representation z_{e_i} and the embedding vector $e_j^{(i)}$ is computed, and the Softmax function is applied to generate a smooth probability distribution

$$p_j^{(i)} = \frac{\exp\left(-\left\|z_{e_i} - e_j^{(i)}\right\|_2^2 \Big/ \tau_{\text{temp}}\right)}{\sum_{l=1}^{K} \exp\left(-\left\|z_{e_i} - e_l^{(i)}\right\|_2^2 \Big/ \tau_{\text{temp}}\right)} \tag{11}$$

where $p_j^{(i)}$ indicates the relevance between z_{e_i} and the discrete vector $e_j^{(i)}$, and τ_{temp} is a temperature hyperparameter that controls the smoothness of the distribution. Using this distribution, the latent representation z_{e_i} is quantized as a weighted average of the discrete vectors in the embedding space

$$z_{e_i}^{\text{vq}} = \sum_{j=1}^{K} p_j^{(i)} \cdot e_j^{(i)}. \tag{12}$$

This method ensures that all discrete vectors are considered during quantization, preventing the "jumping" and gradient breakdown issues associated with hard quantization. It also helps to avoid mode collapse, as each discrete vector has an opportunity to contribute to the quantization process. In addition to the VQ loss, the reconstruction loss and the distribution loss from the VAE process must also be incorporated. The process principle is shown in *step 4* of Fig. 3, which is expressed as

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(z|x)}\left[\log p(x|z)\right] - \mathbb{D}_{\text{KL}}\left[q(z|x)||p(z)\right]. \tag{13}$$

The first term represents the reconstruction loss, $\mathbb{E}_{q(z|x)}[\log p(x|z)]$, which measures the difference between the original data and the reconstructed data produced by the decoder. The second term, the kullback–leibler (KL) divergence loss, $\mathbb{D}_{\text{KL}}[q(z|x)||p(z)]$, quantifies the difference between the learned latent space distribution and the prior distribution. To mitigate the gradient explosion issues caused by large distribution differences in the KL divergence, we use Wasserstein distance as a substitute, allowing for smoother gradient updates. The final optimization objective is then expressed as

$$\mathcal{L}_{\text{VAE}} = \frac{1}{m}\sum_{j=1}^{m}\left[\frac{1}{n}\sum_{i=1}^{n}\left(\tilde{\mathbf{x}}_i^{\text{ER}(j)} - \mathbf{x}_i^{\text{ER}(j)}\right)^2 + \left(f_w\left(x^{\text{ER}(j)}\right) - f_w\left(\text{Decoder}\left(z_{e_j}^{\text{vq}}\right)\right)\right)\right]. \tag{14}$$

Finally, the total optimization objective is the sum of the VAE loss and the VQ loss

$$\mathcal{L}_{\text{total}} = \gamma_{\text{VAE}} \cdot \mathcal{L}_{\text{VAE}} + \gamma_{\text{vq}} \cdot \mathcal{L}_{\text{vq}}. \tag{15}$$

D. FC-MC-SVQ-Based DT-Enhanced Modeling

In DT-based data modeling, challenges such as multimodal complexity, multiscale variations, and nonlinear dependences must be addressed. To tackle these issues,

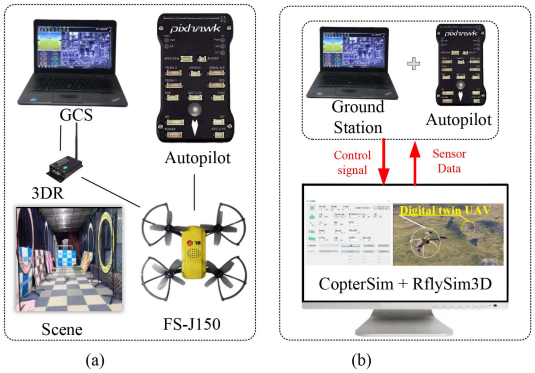


Fig. 4. Experimental configuration of real UAV and DT UAV.

this article presents an enhanced modeling method based on feature clusteringmulti-channel conditional soft vector quantization (FC-MC-SVQ), with the algorithm and training framework detailed in Algorithm 1. The proposed method begins with a condition-aware encoder, which integrates sensor data with external variables. A multichannel decomposition strategy then partitions the feature space into subspaces, capturing local correlations across different modalities. During quantization, a soft quantization mechanism evaluates the similarity between latent variables and subspace codebooks, while a straight-through estimator [41] improves information backpropagation. In the decoding stage, a condition-aware decoder reconstructs the features, minimizing reconstruction error to refine model performance. To further align the reconstructed data with the original distribution, a Wasserstein distance constraint is applied, reducing statistical discrepancies.

III. RESULTS

A. Hardware Configuration

This study utilizes a hybrid method combining real and virtual UAVs, as shown in Fig. 4. The real UAV system is built on the Racer flight controller, an improved version of the open-source Pixhawk. Flight data are transmitted in real-time via a 3DR telemetry module. The UAV operates on an FS-J150 quadcopter frame from Feisi Laboratory,[1] with a ground control station (GCS) featuring an Intel i9-12900H processor (2.5 GHz) for command execution and monitoring. The virtual UAV system mirrors the real UAV in both flight control firmware and GCS configuration. It runs a DT model within a high-precision motion simulator, ensuring its control responses closely match real-world flight dynamics.

To assess the effectiveness of the proposed time-series data generation method, sensor data were collected from two key flight modes: takeoff and hovering. The dataset includes eight valid flight recordings, with six allocated for training (three for takeoff and three for hovering) and two for validation (one for each mode). Data collection focused on the UAV's 3-D motion characteristics. Key measurements include readings from a triaxial accelerometer, gyroscope, and magnetometer, as well as position-filtered data.

B. Experimental Setup

1) *Data Analysis:* Fig. 5(a) shows the distribution of multidimensional time-series sensor data collected from real-world drone flights, along with their filtered counterparts. The dataset includes 12 channels corresponding to accelerometers, gyroscopes, magnetometers, and position estimators. The figure highlights clear variations across channels, such as differences in scale and skewness. These disparities reflect the complexity of modeling heterogeneous sensor data and emphasize the need for methods that can adapt to nonuniform, multiscale distributions. Our feature clustering-based method is designed to address these challenges by capturing structure within diverse data types. Fig. 5(b) presents an example of a typical network input sequence. This input includes synchronized triaxial data from the accelerometer, gyroscope, magnetometer, and position estimator. The plot illustrates distinct temporal behaviors across sensors: for example, accelerometer and gyroscope data often contain short bursts or spikes, while magnetometer readings remain relatively stable, and position data show smooth, periodic trends. These variations in temporal dynamics make it essential for the model to learn time-dependent features across multiple data types effectively. Fig. 5(c) illustrates the residuals between actual sensor values and DT outputs, representing the system's current modeling error. The residuals differ significantly across sensor types, with some showing skewed or nonstationary patterns. These inconsistencies highlight the difficulty of achieving robust and generalized modeling across all channels. In summary, drone sensor data present significant modeling challenges due to its high dimensionality, heterogeneity, and varying temporal and distributional characteristics. Our proposed framework—based on feature clustering, multichannel VQVAE, and soft quantization—offers an effective solution by adapting to these complexities and enabling more accurate and resilient DT-enhanced modeling.

2) *Baseline Models:* To evaluate the performance of the proposed method in generating multimodal and multiscale time-series data for UAVs, we compare it with several state-of-the-art deep generative models. These models represent a range of strategies, including conditional generation, adversarial training, quantized embedding, contrastive learning, and multiscale modeling, enabling a comprehensive assessment of the strengths and limitations of our method.

1) *CVAE (2015) [42]:* A conditional extension of the VAE for feature-controlled generation.
2) *VQVAE (2017) [43]:* Discretizes latent space via codebooks to improve reconstruction fidelity.
3) *WGAN (2017) [44]:* Introduces Wasserstein loss to stabilize adversarial training and avoid mode collapse.

[1][Online]. Available: http://feisilab.com/

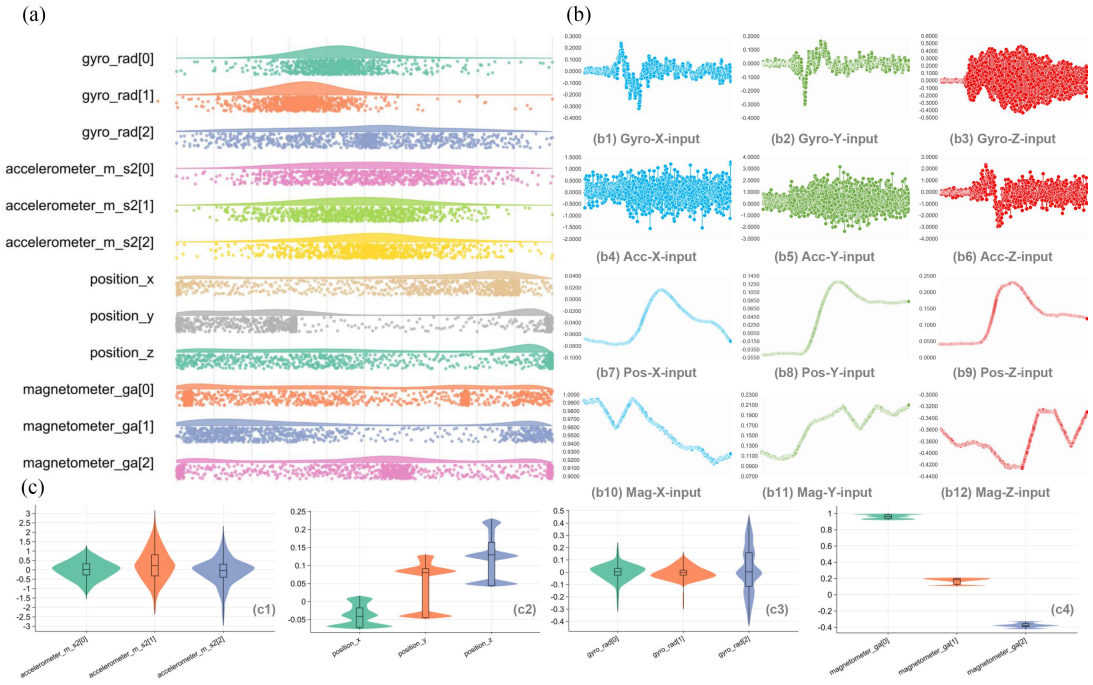


Fig. 5. Feature analysis of real-world drone data and network input sequences. (a) Raincloud plot of multidimensional raw sensor data collected from real drone flights. (b) Schematic illustration of a sampled network input sequence. The input consists of multiple subchannels corresponding to 12-D time-series data, including (X/Y/Z, unit: rad/s), accelerometer (X/Y/Z, unit: m/s^2), position (X, Y, Z, unit: m), and magnetometer (X/Y/Z, unit: Gauss) measurements. Each subchannel represents the residual signal between real-world flight data and the outputs of the DT model. (c) Violin plots of sensor residuals, i.e., the differences between real measurements and unenhanced DT outputs. Subplots include: (c1) Residual distribution of accelerometer signals, (c2) Residual distribution of position signals, (c3) Residual distribution of gyroscope signals, and (c4) Residual distribution of magnetometer signals.

4) *TSGAN (2020) [45]:* Tailored for time series, with layered wasserstein generative adversarial network (WGAN) architecture and temporal constraints.
5) *VQGAN (2021) [46]:* Combines VQ with GAN training to preserve fine details in generated data.
6) *CGAN (2022) [47]:* Injects conditional information into GANs for guided data generation.
7) *MRGAN (2024) [40]:* Introduces modality-aware guidance via pretrained regression networks.
8) *VQWWVAE (2024) [39]:* Fuses VQ with Wasserstein objectives to model complex, multimodal distributions.
9) *TDM (2024) [48]:* Time diffusion model for time-series data generation.
10) *TimeDDPM (2024) [49]:* Time-series augmentation strategy for soft sensing based on the diffusion model.
11) *AutoFilterVQ (ours-extended):* Enhances VQVAE with multiscale adaptive filtering to capture frequency-aware representations.
12) *ContrastVQ (ours-extended):* Incorporates contrastive learning into VQ to distinguish positive and negative latent samples.
13) *MCVQ (ours-extended):* Applies separate quantization to high- and low-frequency channels for multiscale representation learning.

REMARK 2 AutoFilterVQ, ContrastVQ, and MCVQ are included as extended comparisons to the proposed method, using identical strategies and parameter settings to ensure fairness in the experiments. This allows for a direct evaluation of how differences in network architecture influence generation performance under the same generation strategy, providing insight into the specific role of network structure in data generation quality.

REMARK 3 To ensure the fairness of comparison with the model proposed in this article, all baseline models adopt the same training strategy as employed in this study. This includes the parameters listed in Table I , such as "Epochs, Optimizer, Batch size, In/out Dims, Seed, and Device" as well as the data preprocessing and standardization methods.

3) *Evaluation Metrics:* To evaluate the generative performance of each method, we assess two key aspects: error assessment and mode distribution matching. For error assessment, we use two error metrics—root mean squared error (RMSE) and mean absolute error (MAE)—to measure the numerical difference between generated and real data, providing insight into the accuracy of the generation process. For mode distribution matching, we assess the similarity between the distributions of the generated and real data using the Wasserstein distance (WD). This helps verify the model's ability to adapt to different flight modes.

TABLE I
Hyperparameter and Training Configuration Settings

Parameter	Value	Description
C	4	Number of feature clusters used in the clustering-aware module
K	512	Number of embeddings in each codebook for vector quantization
D	128	Dimensionality of the latent embedding vector
τ_{temp}	1.0	Temperature hyperparameter controls distribution smoothness
β	0.25	Weight of the commitment loss term in the quantization objective
γ_{VAE} / γ_{vq}	1.0 / 1.0	Weight hyperparameters of the loss function
α	1×10^{-3}	Initial learning rate for Adam optimizer
Optimizer	Adam	Optimization algorithm and coefficients.
Batch size	4	Number of samples per training batch
In/out Dims	15 / 15	The input and output dimensions of the network
Epochs	250	Total number of training iterations
Seed	42	Random seed for reproducibility
Encoder	$[64, 128]$	Two-layer encoder with LSTM
Decoder	$[128, 64, 32, 15]$	Four-layer decoder with LSTM
Device	RTX 3060 GPU	Hardware used for model training and evaluation

TABLE II
Comparison of Models on Validation Dataset

Models	*Modality 1*									*Modality 2*								
	RMSE			MAE			WD			RMSE			MAE			WD		
	100%	*50%*	*25%*	*100%*	*50%*	*25%*	*100%*	*50%*	*25%*	*100%*	*50%*	*25%*	*100%*	*50%*	*25%*	*100%*	*50%*	*25%*
CVAE	0.8748	0.8746	0.8754	0.3687	0.3686	0.3684	0.2876	0.2875	0.2871	1.2954	1.3120	1.3134	0.4761	0.4781	0.4783	0.3738	0.3737	0.3716
VQVAE	0.2626	0.2628	0.2634	0.1826	0.1828	0.1833	0.1192	0.1193	0.1200	0.4239	0.4453	0.4501	0.2265	0.2309	0.2339	0.1250	0.1241	0.1270
WGAN	0.2137	0.2179	0.2162	0.2113	0.2131	0.2118	0.2066	0.2083	0.2070	0.3221	0.3362	0.3509	0.2664	0.2696	0.2776	0.2549	0.2579	0.2663
TSGAN	3.8072	3.8089	3.8035	0.7774	0.7776	0.7770	0.6628	0.6629	0.6626	5.8858	5.8931	5.9093	0.9648	0.9652	0.9676	0.8197	0.8177	0.8176
VQGAN	0.1170	0.1172	0.1175	0.1400	0.1402	0.1400	0.1119	0.1118	0.1118	0.1945	0.2015	0.2085	0.1792	0.1813	0.1841	0.1280	0.1280	0.1301
CGAN	0.1475	0.1480	0.1471	0.1709	0.1709	0.1704	0.1660	0.1660	0.1655	0.3323	0.3475	0.3555	0.2512	0.2555	0.2587	0.2421	0.2460	0.2493
MRGAN	23.863	23.817	23.779	2.3903	2.3881	2.3869	2.3870	2.3849	2.3836	32.988	33.121	33.125	3.0759	3.0813	3.0798	3.0728	3.0782	3.0766
TDM	0.4884	0.4879	0.4845	0.2953	0.2953	0.2942	0.1495	0.1498	0.1503	0.8078	0.8253	0.8455	0.3794	0.3827	0.3863	0.2052	0.2018	0.2020
TimeDDPM	0.1888	0.1903	0.1910	0.1891	0.1896	0.1901	0.0938	0.0943	0.0950	0.4507	0.4522	0.4527	0.2942	0.2942	0.2935	0.2172	0.2169	0.2155
VQWWVAE	0.0851	0.0856	0.0865	0.1152	0.1156	0.1161	0.0641	0.0640	0.0641	0.1519	0.1587	0.1680	0.1498	0.1527	0.1571	0.0773	0.0791	0.0815
AutoFilterVQ	0.0050	0.0050	0.0046	0.0292	0.0292	0.0286	0.0219	0.0219	0.0215	0.0211	0.0233	0.0237	0.0555	0.0580	0.0587	0.0335	0.0345	0.0349
ContrastVQ	0.5596	0.5527	0.4757	0.2407	0.2407	0.2332	0.1009	0.1007	0.0929	1.2291	1.3194	1.3425	0.3598	0.3741	0.3811	0.1718	0.1802	0.1827
MCVQ	0.0237	0.0237	0.0236	0.0555	0.0556	0.0556	0.0335	0.0334	0.0334	0.0393	0.0405	0.0406	0.0746	0.0758	0.0764	0.0321	0.0327	0.0328
Proposed	**0.0012**	**0.0013**	**0.0012**	**0.0165**	**0.0166**	**0.0164**	**0.0137**	**0.0138**	**0.0136**	**0.0047**	**0.0050**	**0.0050**	**0.0315**	**0.0323**	**0.0329**	**0.0231**	**0.0236**	**0.0240**

The hyperparameter settings during the experiment are shown in Table I.

C. Experimental Results

1) *Comparison of Model Performance:* To thoroughly assess the performance of different generative methods, we compare RMSE, MAE, and WD on the validation dataset. The results are presented in Table II. Among the evaluated methods, mode-related generative adversarial network (MRGAN) performs the worst due to its strong dependence on regression-based fitting. Given the high-dimensional and nonlinear nature of time-series data in this task, it struggles to capture complex dynamic patterns effectively. In contrast, autoencoder-based models (e.g., vector-quantized weighted-wasserstein variational autoencoder (VQWWVAE) and MCVQ) outperform GAN-based methods [e.g., time-series generative adversarial network (TSGAN), vector-quantized generative adversarial network (VQGAN), and conditional generative adversarial network (CGAN)] by leveraging superior nonlinear feature extraction, allowing for more accurate modeling of time-series distributions. The proposed method further enhances the autoencoder structure, significantly reducing generation errors and achieving superior results across multiple metrics. A broader extended comparison highlights that MCVQ and AutoFilterVQ improve nonlinear data modeling through multiscale frequency decomposition. Meanwhile, the proposed method employs a more efficient feature fusion and reconstruction strategy, further enhancing data quality. It excels in feature representation, modality alignment, and stability. Regardless of data availability, the proposed method demonstrates high stability and robustness, performing consistently well under 100%, 50%, and 25% data conditions.

2) *Comparison of Reconstruction Performance:* To further evaluate the reconstruction capabilities of different methods in UAV time-series data generation, we randomly select a sample from the validation set and compare the generated results. As shown in Fig. 6, the figure illustrates the reconstruction performance of different methods across accelerometer (X/Y/Z), gyroscope (X/Y/Z), magnetometer (X/Y/Z), and position data (X/Y/Z). The proposed method achieves superior reconstruction quality across all modalities, effectively capturing the dynamic variations in time-series data and accurately predicting its nonlinear structures. In contrast, other methods exhibit varying degrees of deviation. VQWWVAE demonstrates stable short-term predictions but accumulates errors over longer sequences, leading to a gradual divergence from the true trajectory. VQGAN captures some nonlinear data

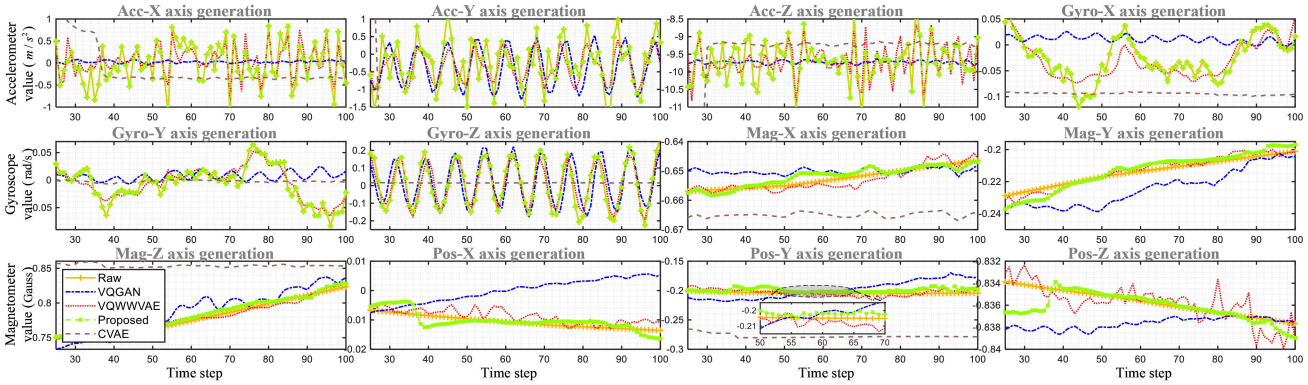


Fig. 6. Comparison of reconstruction performance across different methods for UAV time-series data generation. The figure presents 12 dimensions including accelerometer (X/Y/Z, unit: m/s^2), gyroscope (X/Y/Z, unit: rad/s), magnetometer (X/Y/Z, unit: Gauss), and positional information (X/Y/Z), generated using the proposed method, VQGAN, VQWWVAE, and CVAE. Given that the full comparison involves 12 methods, displaying all curves simultaneously may compromise visual clarity and hinder direct comparisons. Therefore, this figure selectively showcases representative approaches to more clearly illustrate differences in reconstruction accuracy and temporal consistency among various generative strategies. For a more comprehensive performance comparison, see Figs. 7 and 8.

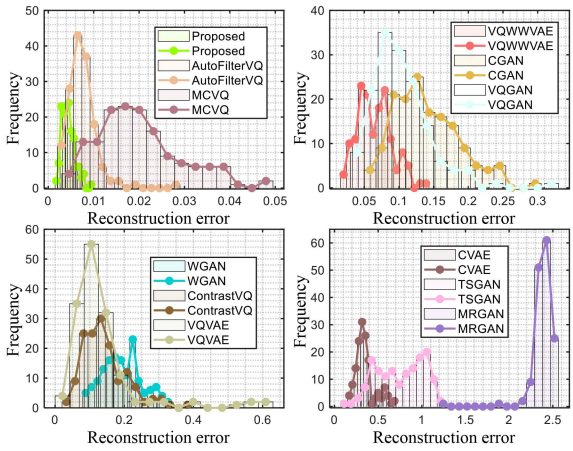


Fig. 7. Histogram comparison of reconstruction errors across different methods for UAV time-series data generation. The histogram presents the average reconstruction error for 12 methods, calculated as the mean absolute error across 12 data dimensions, including the accelerometer (X/Y/Z), gyroscope (X/Y/Z), magnetometer (X/Y/Z), and positional information (X/Y/Z).

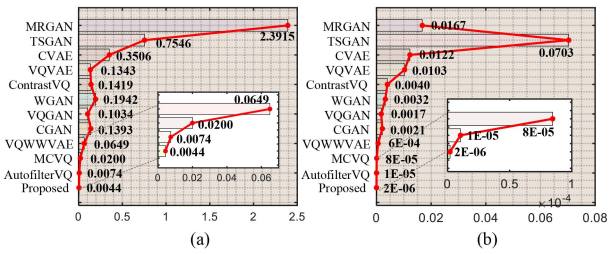


Fig. 8. Comparison of the mean and variance of reconstruction errors across different methods for UAV time-series data generation. The errors are computed as the mean and variance of the absolute errors across 12 data dimensions, including the accelerometer (X/Y/Z), gyroscope (X/Y/Z), magnetometer (X/Y/Z), and positional information (X/Y/Z).

characteristics and can approximate general trends but struggles with linear patterns, reducing reconstruction accuracy in certain modalities. conditional variational autoencoder (CVAE), on the other hand, suffers from significant fluctuations in most cases, with some data points deviating substantially from ground truth, indicating its limited capability in modeling complex time-series structures and learning stable temporal dynamics.

The quantitative results in Figs. 7 and 8 further highlight these differences. Fig. 7 shows the error distribution, where the proposed method achieves the smallest deviation from the ground truth, with a maximum error below 0.01 and a concentrated distribution, demonstrating high consistency with real data. In contrast, other methods show greater error variation, with MRGAN reaching a maximum error of 2.5, reflecting its poor adaptability to complex time-series patterns. Fig. 8 provides a statistical analysis of the mean and variance of errors, showing that the proposed method achieves both the lowest mean error and the smallest variance, ensuring stable data generation while mitigating mode collapse and data drift. Overall, the proposed method delivers higher reconstruction accuracy, more precise temporal predictions, and stronger generalization in UAV time-series data generation. It effectively captures the dynamic characteristics of UAV data, producing high-quality and stable sequences.

3) *Experimental Verification of Real Flight:* To evaluate the effectiveness and adaptability of the proposed method in real UAV flight scenarios, we conducted validation experiments on Hovering and Takeoff missions, further extending the assessment to Tour and Forward missions. This comprehensive evaluation examines the method's reliability and performance across various flight conditions. The experimental results are presented in Fig. 9. For Hovering and Takeoff missions, significant disparities were noted between the DT simulation data and real flight data across various sensor readings, encompassing accelerometer, gyroscope, magnetometer, and position data. These differences primarily result from modeling inaccuracies and limitations in capturing nonlinear features within simulation environments. In contrast, the proposed method effectively

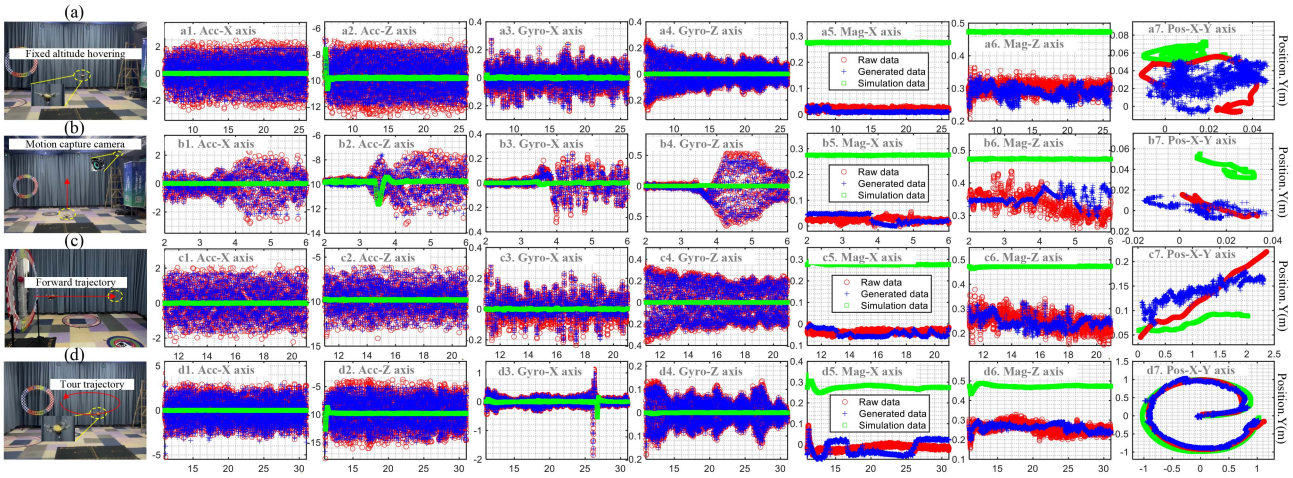


Fig. 9. Multidimensional performance comparison under different real-world flight mission experiments. The experiment includes four distinct missions under both static and dynamic conditions: (a) Hovering, (b) Takeoff, (c) Forward flight, and (d) Tour. For each mission, the generated data are visualized across multiple sensor dimensions, including accelerometer (Acc-X, Acc-Z, unit: m/s^2) in (a)1–(d)1 and (a)2–(d)2, gyroscope (Gyro-X, Gyro-Z, unit: rad/s) in (a)3–(d)3 and (a)4–(d)4, magnetometer (Mag-X, Mag-Z, unit: Gauss) in (a)5–(d)5 and (a)6–(d)6, and position (Pos-X-Y, unit: m) in (a)7–(d)7. To ensure the stability and reliability of the experimental data, real flight data were collected using an indoor motion capture system.

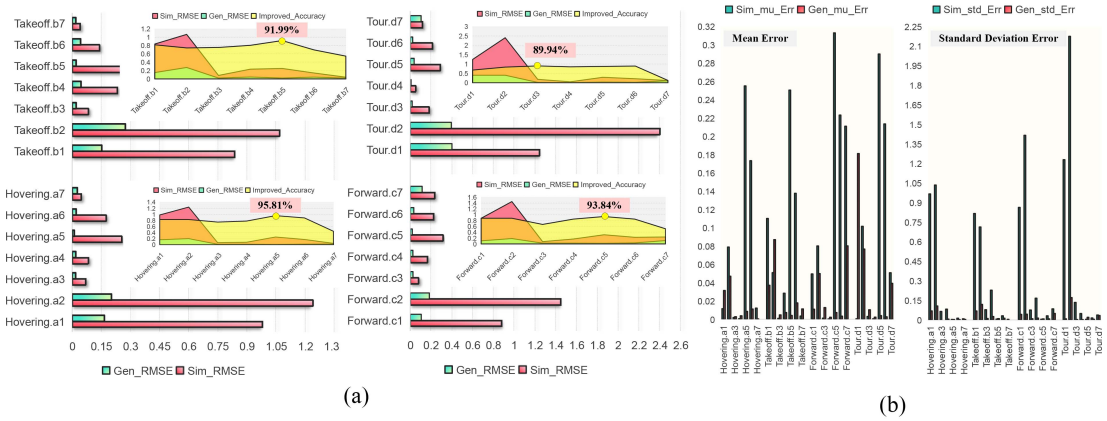


Fig. 10. Performance comparison results. (a) Percentage improvement in data accuracy across different missions and sensor modalities. Gen_RMSE and Sim_RMSE represent the RMSE between the generated/simulated data and real data, respectively. Higher improvement indicates better alignment between generated data and real-world behavior. (b) Comparison of distributional consistency across tasks. Gen_mu_Err and Sim_mu_Err denote the mean error between generated/simulated and real data, while Gen_std_Err and Sim_std_Err denote the discrepancy in standard deviation. These metrics evaluate the ability of models to compare the statistical properties of sensor data.

reduces these discrepancies by incorporating precise feature extraction and reconstruction techniques. The generated data closely matches real flight data in trend, fluctuation, and nonlinear characteristics, demonstrating strong temporal modeling and generalization performance. Similarly, in Tour and Forward missions, the method consistently produces accurate and stable synthetic data, successfully capturing complex flight trajectories and dynamic patterns. It also excels in long-sequence modeling and predicting nonlinear variations.

Fig. 10(a) further quantifies accuracy improvements across different sensor modalities for each mission. The results show that the proposed method significantly reduces generated errors, with the highest accuracy improvements reaching 95.81% for Hovering and 93.84% for Forward, highlighting its robust dynamic response modeling and nonlinear feature reconstruction. Despite the increased complexity of Tour task trajectories, the method maintains high stability and adaptability, demonstrating its reliability across multiple UAV missions. The data distribution comparison in Fig. 10(b) also confirms this conclusion. Overall, the proposed method accurately captures UAV dynamics, generating high-quality, stable, and generalizable synthetic data. These findings confirm its effectiveness and superiority for real-world UAV applications.

To evaluate model performance in real-world scenarios, we applied the four top-performing methods from Table II—Proposed, AutoFilterVQ, VQWWVAE, and MCVQ—to actual UAV missions. Fig. 11 presents the quantitative comparison results, which cover both static (Hovering) and dynamic (Takeoff, Forward, and Tour) missions. Performance is assessed using RMSE, MAE, and WD. The results show that the proposed method consistently outperforms baseline models across all missions. In

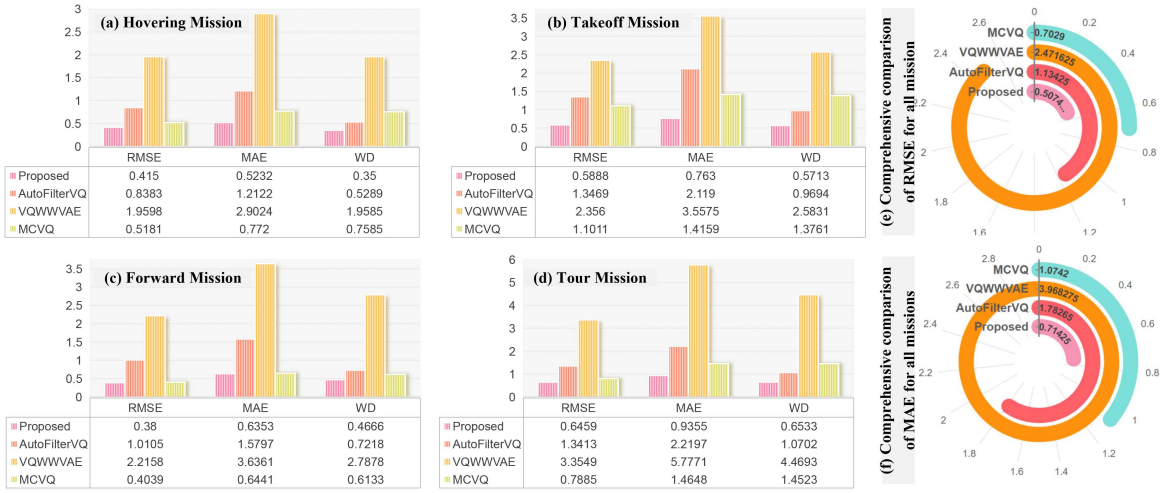


Fig. 11. Quantitative performance comparison of different models, including the proposed model, AutoFilterVQ, VQWWVAE, and MCVQ, across all missions. Evaluation metrics include RMSE, MAE, and WD, providing a comprehensive assessment of reconstruction accuracy and distribution fidelity across methods.

more complex dynamic scenarios, such as Forward and Tour, it achieves lower errors and higher generation accuracy. Despite being trained only on Hovering and Takeoff missions, the model effectively generates high-quality data for Forward and Tour missions, preserving distribution consistency. This demonstrates its strong generalization and task-transferability. Overall, these findings validate the FC-MC-SVQ mechanism as an effective method for DT-based modeling, providing reliable data support for UAV perception and mission execution in unknown environments.

4) *Ablation Study:* To assess the effectiveness of the proposed method, we conduct an ablation study on the key components of the FC-MC-SVQ framework, evaluating their impact on overall performance. Specifically, we systematically remove the FC module (denoted as Without_FC in Fig. 12(b), where feature clustering categories are reduced from four to two), the multichannel conditional (MC) module (Without_MC), and the MC module (Without_SVQ). Experiments are performed on both the Hovering and Takeoff datasets, using validation sets with 100%, 50%, and 25% of the data to thoroughly examine each module's contribution. The results are presented in Fig. 12(b).

The experimental results reveal that removing the FC module has a relatively small effect on performance, while eliminating the MC module leads to a significant decline—by an order of magnitude. Specifically, the MC module is essential for organizing multiscale and multimodal data into separate channels based on feature clustering, allowing for more effective extraction of related features. Without it, the model struggles to capture feature relationships, reducing information integration and overall performance. This highlights the MC module as a critical component of the framework. In addition, the SVQ module also plays a key role by replacing traditional hard quantization, which helps reduce quantization errors and improve generalization. While removing SVQ does not cause a dramatic drop in performance, it does lead to noticeable degradation in model accuracy, confirming its role in minimizing information loss and enhancing robustness. It is also worth noting that the number of feature clustering categories in the FC module affects model performance. Reducing the number of categories from four to two decreases feature separation accuracy, preventing the model from effectively capturing key characteristics. However, this impact is less pronounced than that of the MC module. These results suggest that choosing an appropriate number of clustering categories is essential to balancing computational efficiency and modeling accuracy. Overall, the ablation study strongly supports the design and effectiveness of the proposed FC-MC-SVQ framework.

We further conducted a quantitative ablation study to assess the contribution of individual model components to reconstruction performance across different sensor types, as shown in Fig. 12(a), (c), (d), and (e). For sensors characterized by high-frequency variation—such as accelerometers and gyroscopes—the complete model consistently delivered the most accurate reconstructions under both static (Hovering) and dynamic (Takeoff, Forward, and Tour) flight conditions. This demonstrates the model's effectiveness in capturing detailed temporal patterns. In contrast, removing components such as FC, MC, or SVQ led to a marked decrease in performance, particularly in high-dynamic scenarios, where the ability to recover fine signal details was noticeably reduced. These findings underscore the importance of these modules in modeling complex, dynamic behaviors. In comparison, for sensors with more stable outputs– - such as magnetometers and position data—the differences in reconstruction performance among model variants were relatively minor across all flight conditions. This suggests that for low-frequency or less variable signals, the model's architectural complexity has a limited impact on reconstruction accuracy. Overall, the ablation results confirm the essential role of each

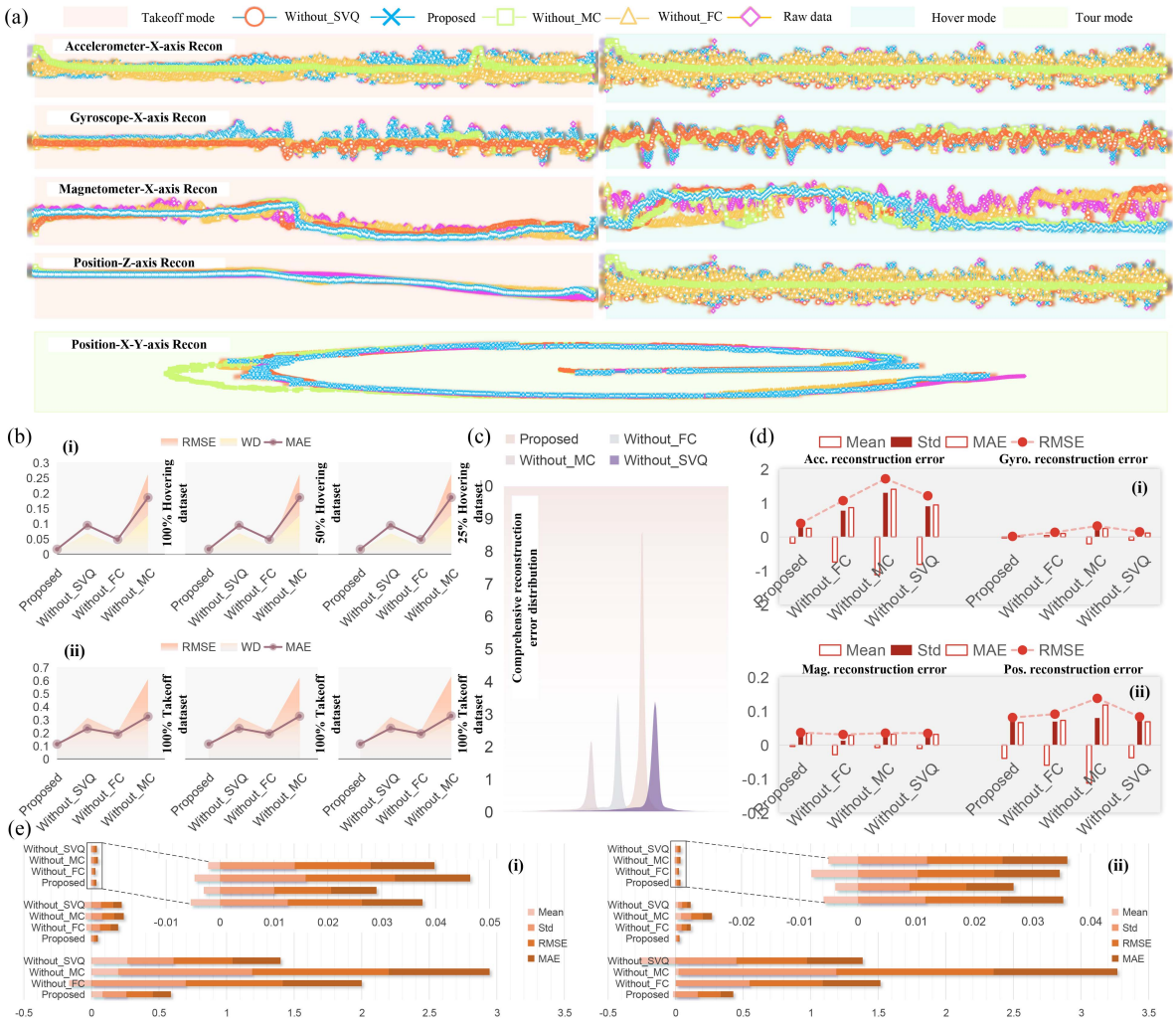


Fig. 12. Comparative results of the ablation study. (a) Reconstruction performance under different flight modes. This subplot illustrates the reconstruction results of four sensor modalities—accelerometer, gyroscope, magnetometer, and position data—under various UAV flight modes, using different ablation configurations (Proposed, Without_FC, Without_MC, and Without_SVQ). (b) Performance evaluation under different data scales: (i) In the Hovering mode, the performance of each method is assessed using datasets of varying sizes (100%, 50%, and 25%) based on RMSE, WD, and MAE metrics, aiming to evaluate robustness under limited data conditions, and (ii) a similar evaluation is conducted for the Takeoff mode to investigate method stability and adaptability under dynamic flight scenarios. (c) Distributional comparison of reconstruction errors across flight modes. This part presents the overall error distributions of different methods across all flight modes using kernel density estimation. The plots offer insight into the global reconstruction stability and error characteristics of each method. (d) Statistical error analysis under the Tour mode: (i) Comparison of reconstruction errors (mean, standard deviation, MAE, and RMSE) between the accelerometer and gyroscope, and (ii) performance comparison for the magnetometer and position data. (e) Error comparison in Hovering and Takeoff modes: (i) In the Hovering mode, the performance of different methods is compared for gyroscope, magnetometer, and accelerometer data, and (ii) in the Takeoff mode, the same three sensors are analyzed to assess method effectiveness in rapidly changing flight states.

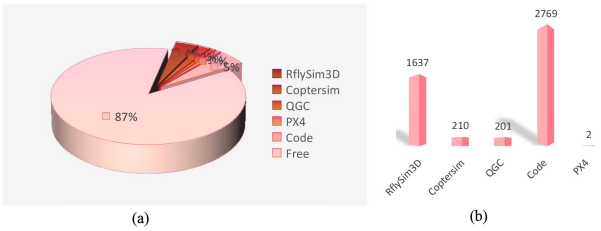
module in improving the model's capability, especially under dynamic conditions, and highlight the proposed framework's robustness and adaptability in diverse UAV flight environments.

TABLE III
Small Sample Ratio Experiment

Model	*Modality 1*			*Modality 2*		
	RMSE	MAE	WD	RMSE	MAE	WD
10%-Model	0.0031	0.0276	0.0227	0.0073	0.0415	0.0310
30%-Model	0.0016	0.0204	0.0180	0.0044	0.0313	0.0235
60%-Model	0.0018	0.0227	0.0192	0.0053	0.0391	0.0309
100%-Model	0.0012	0.0165	0.0137	0.0047	0.0315	0.0231

D. Discuss

As shown in Table III, we analyzed the model performance under varying training data volumes. The results indicate that, overall, the model exhibits a controlled degradation trend as the training data decreases: when the data volume decreases from 100% to 60%, all metrics show only slight increases, and the model still maintains good reconstruction accuracy; as the data further decreases to 30% and 10%, the errors increase more noticeably, but the model retains basic pattern reconstruction capability. Notably, within the 30%–60% data range, the RMSE, MAE,



(a) (b)

Fig. 13. Performance statistics for edge device deployments. (a) CPU usage performance (%) statistics of edge device deployment process. (b) CPU memory usage (Mb) statistics of edge device deployment process.

TABLE IV
Model Inference Resource Usage

Infer. Time (ms)	GPU (%)	GPU Memory (MB)	CPU Memory (MB)
165 ± 5	78 ± 3	2094	2741 ± 12

TABLE V
Comparison of Ablation Experiments on Loss Function Weights

Model	*Modality 1*			*Modality 2*		
	RMSE	MAE	WD	RMSE	MAE	WD
$\gamma_{\text{vq}} = 0.1$	0.0020	0.0194	0.0149	0.0063	0.0351	0.0233
$\gamma_{\text{vq}} = 0.5$	0.0019	0.0195	0.0158	0.0063	0.0360	0.0266
$\gamma_{\text{vq}} = 0.9$	0.0014	0.0181	0.0149	0.0049	0.0334	0.0243
$\gamma_{\text{VAE}} = 0.1$	0.2488	0.1832	0.1400	0.3969	0.2261	0.1714
$\gamma_{\text{VAE}} = 0.5$	0.0122	0.0416	0.0354	0.0061	0.0397	0.0302
$\gamma_{\text{VAE}} = 0.9$	0.0014	0.0176	0.0142	0.0058	0.0345	0.0242

and WD metrics show slight increases. This is primarily because the dataset consists of raw sensor data without denoising, causing noise and abnormal readings to accumulate as the sample size increases, thereby introducing stronger interference to the model. Overall, this phenomenon reflects the model's true behavior under noisy, raw data conditions.

To evaluate the feasibility of deploying the proposed model on edge devices, experimental tests were conducted, and the results are illustrated in Fig. 13 and Table IV. It can be observed from the experimental results that the average inference time of the model on edge devices is approximately 165 ms, which falls within the typical control cycle of UAVs, and thus, satisfies the real-time processing requirement. Meanwhile, the GPU memory consumption is about 2 GB, and the CPU memory usage remains around 2.7 GB. The model runs stably throughout the entire run, but there is room for further optimization. These findings indicate that the method proposed in this study possesses high deployment feasibility on edge terminals. In future work, we plan to further deploy and test the model on onboard hardware (e.g., NVIDIA Jetson Orin Nano) and explore techniques such as model pruning and dynamic quantization to further reduce computational complexity, thereby enabling onboard deployment for actual flight tasks.

We further investigated the impact of varying the weighting coefficients of different loss components, as summarized in Table V. The results indicate that changes in the VQ-Loss weight have only a marginal effect on overall performance, primarily because this term serves as a structural regularization mechanism in the latent feature space rather than directly influencing reconstruction accuracy. In contrast, the VAE-Loss weight directly governs the model's reconstruction capability and dynamic consistency along the temporal dimension. Consequently, the model performance is more sensitive to variations in γ_{VAE}, while remaining relatively stable with respect to changes in γ_{vq}.

E. Source Code and Data

To ensure reproducibility, the complete dataset, source code, training scripts, and pretrained models used in this work have been released online,[2] which provides the following:

1) hybrid DT datasets containing real flight and simulation data;
2) UAV DT modeling and control code;
3) network implementations and training pipelines;
4) pretrained model weights and configuration files.

IV. CONCLUSION

DT technology is becoming a key enabler for advancing UAV systems in the low-altitude economy. It plays a vital role in intelligent perception and collaborative decision making. However, conventional modeling methods often fall short in real-world applications due to the complexity of multisource data, varying data scales, and nonlinear system behaviors. These challenges make it difficult to build reliable, accurate, and robust models for UAVs operating in dynamic environments. To overcome these limitations, we propose a novel UAV DT-enhanced modeling method based on feature clustering, multichannel VQVAE, and a soft quantization mechanism. By incorporating a soft vector quantization strategy and a feature-space clustering mechanism, the proposed method enables structured modeling of multiscale, multimodal sensor data. It effectively captures and reconstructs nonlinear coupled features across different scales while enhancing robustness to sensor noise and dynamic variations. Building on this, we construct a hybrid modeling framework that integrates both physical priors and data-driven mechanisms, facilitating real-time perception and dynamic correction of system states and sensor errors throughout the UAV's flight process.

Our experimental results confirm the advantages of this method in the following three key areas.

1) In model performance, the proposed method outperforms mainstream generative models in both reconstruction accuracy and consistency under challenging conditions. It also better preserves the temporal characteristics of flight data, which are crucial for real-time applications.
2) In engineering applicability, when applied to various UAV missions—such as hovering, takeoff, tour, and forward flight—our method significantly enhances the predictive power of the DT system. The highest performance improvements of

[2][Online]. Available: https://github.com/RflySim/RflySimDT

95.81%, 91.99%, 89.94%, and 93.84%, respectively, highlight its adaptability and effectiveness across diverse operational scenarios.

3) In technical ecosystem, we also developed and released an open-source hardware-in-the-loop DT simulation platform, fully compatible with popular UAV software stacks such as PX4 and ROS. This platform enables real-time interaction between digital and physical UAVs, laying the groundwork for future self-evolving DT systems. Furthermore, the multimodal flight dataset used in this study has been made publicly available to support further research in soft sensing and DT-based modeling.

The core contribution of this study lies in the paradigm of physical-to-digital mapping, wherein a generative hybrid modeling framework is employed to enhance the fidelity of the DT's representation of multimodal UAV sensor data. It is essential to explicitly clarify that the corrected outputs of the DT are not currently used as safety-critical commands to directly control the physical UAV. Instead, they serve as augmented perceptual information to support state monitoring, anomaly detection, and decision-making processes.

We fully acknowledge that realizing closed-loop, safety-critical applications based on DTs constitutes a system-level challenge, involving fault diagnosis, safety certification, and control assurance. This article, therefore, focuses on establishing the trustworthiness of the model, a foundational prerequisite akin to a high-precision navigation map in autonomous driving, without which advanced functionalities cannot operate safely. Building upon this foundation, we envision a three-layer technical evolution path: The current work achieves the physical-to-digital stage, forming the cornerstone of the framework. Future research will explore the digital-to-physical stage, enabling virtual redundancy control under fault conditions. The long-term goal is to realize digital–physical collaboration, facilitating safe and intelligent decision-making through dynamic interaction between real and virtual agents. This study provides the theoretical and methodological groundwork for this progressive roadmap, while future work will focus on addressing challenges such as online adaptive learning and real-time reliability assurance.

REFERENCES

[1] R. Mon-Williams, G. Li, L. Ran, W.-Q. Du, and C. G. Lucas, "Embodied large language models enable robots to complete complex tasks in unpredictable environments," *Nature Mach. Intell.*, vol. 7, no. 4, pp. 592–601, Apr. 2025.

[2] Y. L. Tian et al., "UAVs meet LLMs: Overviews and perspectives towards agentic low-altitude mobility," *Inf. Fusion*, vol. 122, 2025, Art. no. 103158.

[3] J. Xu, Q. Y. Sun, Q. L. Han, and Y. Tang, "When embodied AI meets industry 5.0: Human-centered smart manufacturing," *IEEE/CAA J. Automatica Sinica*, vol. 12, no. 3, pp. 485–501, Mar. 2025.

[4] Y. J. Hu et al., "Industrial Internet of Things intelligence empowering smart manufacturing: A literature review," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 19143–19167, Jun. 2024.

[5] S. Y. Yang, D. F. Lin, S. M. He, I. Hussain, and L. Seneviratne, "Aerial swarm search for GNSS-denied maritime surveillance," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 3, pp. 3442–3453, Jun. 2024.

[6] T. Jin et al., "Triboelectric nanogenerator sensors for soft robotics aiming at digital twin applications," *Nature Commun.*, vol. 11, no. 1, 2020, Art. no. 5381.

[7] S. Kim and S. Heo, "An agricultural digital twin for mandarins demonstrates the potential for individualized agriculture," *Nature Commun.*, vol. 15, no. 1, 2024, Art. no. 1561.

[8] Z. H. Lyu, J. K. Guo, R. Lou, and H. Lv, "Artificial intelligence based spacecraft resilience optimization in space informatics digital twins," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 61, no. 2, pp. 1834–1847, Apr. 2025.

[9] C. C. Peng and Y. H. Chen, "Fixed-wing unmanned aerial vehicle rotary engine anomaly detection via online digital twin methods," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 741–758, Feb. 2024.

[10] L. Zhao, C. Wang, K. Zhao, D. Tarchi, S. Wan, and N. Kumar, "Interlink: A digital twin-assisted storage strategy for satellite-terrestrial networks," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 3746–3759, Oct. 2022.

[11] X. Zhang et al., "A multi-level digital twin construction method of assembly line based on hybrid worker digital twin models," *Adv. Eng. Inform.*, vol. 62, 2024, Art. no. 102597.

[12] F. Tao, B. Xiao, J. F. Cheng, Q. L. Qi, and P. Ji, "Digital twin modeling," *J. Manuf. Syst.*, vol. 64, pp. 372–389, 2022.

[13] Y. Liu, J. Feng, J. M. Lu, and S. Y. Zhou, "A review of digital twin capabilities, technologies, and applications based on the maturity model," *Adv. Eng. Inform.*, vol. 62, 2024, Art. no. 102592.

[14] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—Streamlining simulation-based systems engineering for industry 4.0," *IEEE Trans. Ind. Inform.*, vol. 14, no. 4, pp. 1722–1731, Apr. 2018.

[15] N. Muraleedharan et al., "Modelling and simulation of UAV systems," *Imag. Sens. Unmanned Aircr. Syst.*, vol. 1, pp. 101–121, 2020.

[16] L. Z. Zeng, X. W. Liao, Z. F. Ma, B. P. Xiong, H. Jiang, and Z. Chen, "Three-dimensional UAV-to-UAV channels: Modeling, simulation, and capacity analysis," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10054–10068, Mar. 2024.

[17] K. Guo, Z. S. Ye, D. T. Liu, and X. Y. Peng, "UAV flight control sensing enhancement with a data-driven adaptive fusion model," *Rel. Eng. Syst. Saf.*, vol. 213, 2021, Art. no. 107654.

[18] P. Freeman, R. Pandita, N. Srivastava, and G. J. Balas, "Model-based and data-driven fault detection performance for a small UAV," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 4, pp. 1300–1309, Aug. 2013.

[19] J. H. Wang, Y. Zhang, and R. A. Ramirez-Mendoza, "Direct data driven scheme for UAV flight control," *IEEE Access*, vol. 10, pp. 108241–108250, 2022.

[20] P. K. Huynh, "Knowledge integration in domain-informed machine learning and multi-scale modeling of nonlinear dynamics in complex systems," Ph.D. dissertation, Dept. of Industrial and Management Systems Engineering, , University of South Florida, South Florida, Tampa, FL, USA, 2023.

[21] S. B. Cheng et al., "Machine learning for modelling unstructured grid data in computational physics: A review," *Inf. Fusion*, vol. 123, 2025, Art. no. 103255.

[22] J. E. Santos, Z. R. Fox, A. Mohan, D. O'Malley, H. Viswanathan, and N. Lubbers, "Development of the Senseiver for efficient field reconstruction from sparse observations," *Nature Mach. Intell.*, vol. 5, no. 11, pp. 1317–1325, Nov. 2023.

[23] J. Y. Zhan, Z. B. Ma, and L. G. Zhang, "Data-driven modeling and distributed predictive control of mixed vehicle platoons," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 572–582, Jan. 2023.

[24] S. Yin, S. X. Ding, X. C. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[25] R. H. Kang et al., "Digital twin modeling of the robotic gluing system for predicting the quality of glue lines and optimizing gluing parameters," *J. Manuf. Syst.*, vol. 80, pp. 1074–1092, 2025.
[26] M. Waseem, C. B. Tan, S. C. Oh, J. Arinez, and Q. Chang, "Machine learning-enhanced digital twins for predictive analytics in battery pack assembly," *J. Manuf. Syst.*, vol. 80, pp. 344–355, 2025.
[27] X. J. Liu, C. X. Wang, F. X. Wang, X. L. Qiu, F. Y. Feng, and Y. Sun, "A generic digital twin model construction strategy for cross-field implementations with comprehensiveness, operability and scalability," *J. Manuf. Syst.*, vol. 80, pp. 366–379, 2025.
[28] Y. Q. Chen, Y. W. Liu, Y. R. Meng, S. H. Yu, and Y. Zhuang, "System modeling and simulation of an unmanned aerial underwater vehicle," *J. Mar. Sci. Eng.*, vol. 7, no. 12, 2019, Art. no. 444.
[29] S. Gebreyohannes, A. Karimoddini, and A. Homaifar, "Applying model-based systems engineering to the development of a test and evaluation tool for unmanned autonomous systems," in *Proc. IEEE Int. Syst. Conf.*, 2020, pp. 1–7.
[30] C. Yang et al., "Digital twin-driven fault diagnosis method for composite faults by combining virtual and real data," *J. Ind. Inf. Integration*, vol. 33, 2023, Art. no. 100469.
[31] H. H. Tao, P. Jia, X. Y. Wang, X. Chen, and L. Q. Wang, "A digital twin-based fault diagnostic method for subsea control systems," *Measurement*, vol. 221, 2023, Art. no. 113461.
[32] Q. W. Li et al., "A multimodal data generation method for imbalanced classification with dual-discriminator constrained diffusion model and adaptive sample selection strategy," *Inf. Fusion*, vol. 117, 2025, Art. no. 102843.
[33] A. Bouman, "Autonomous mission-driven robots in extreme environments," Ph.D. dissertation, Dept. of Mechanical and Civil Engineering, , California Institute of Technology, Pasadena, CA, USA, 2022.
[34] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.
[35] I. Goodfellow et al., "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, 2020.
[36] X. Y. Jiang and Z. Q. Ge, "Improving the performance of just-in-time learning-based soft sensor through data augmentation," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13716–13726, Dec. 2022.
[37] G. C. Zhao, C. H. Zhang, B. Duan, Y. L. Shang, Y. Z. Kang, and R. Zhu, "State-of-health estimation with anomalous aging indicator detection of lithium-ion batteries using regression generative adversarial network," *IEEE Trans. Ind. Electron.*, vol. 70, no. 3, pp. 2685–2695, Mar. 2023.
[38] 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 Trans. Ind. Electron.*, vol. 71, no. 10, pp. 13288–13298, Oct. 2024.
[39] X. He, T. Liu, Y. Zhang, and R. Xie, "Soft sensor modeling based on vector-quantized weighted-Wasserstein VAE for polyester polymerization process," *IEEE Trans. Ind. Inform.*, vol. 20, no. 9, pp. 11338–11347, Sep. 2024.
[40] X. Y. Li, Q. X. Zhu, and Y. L. He, "Data mode-related generative adversarial network for industrial soft sensor application," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 4198–4205, Mar. 2024.
[41] M. Y. Huh, B. Cheung, P. Agrawal, and P. Isola, "Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks," in *Proc. 40th Int. Conf. Mach. Learn.*, 2023, pp. 14096–14113.
[42] K. Sohn, H. Lee, and X. C. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.
[43] A. V. D. Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6309–6318.
[44] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[45] K. E. Smith and A. O. Smith, "Conditional GAN for timeseries generation," 2020, *arXiv:2006.16477*.
[46] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12873–12883.
[47] Y. Chen, D. J. Kempton, A. Ahmadzadeh, J. Z. Wen, A. L. Ji, and R. A. Angryk, "CGAN-based synthetic multivariate time-series generation: A solution to data scarcity in solar flare forecasting," *Neural Comput. Appl.*, vol. 34, no. 16, pp. 13339–13353, 2022.
[48] C. Zheng et al., "A novel time diffusion model for industrial time series data generation," in *Proc. IEEE 13th Data Driven Control Learn. Syst. Conf.*, 2024, pp. 894–899.
[49] Y. Dai, C. Yang, K. X. Liu, A. P. Liu, and Y. Liu, "TimeDDPM: Time series augmentation strategy for industrial soft sensing," *IEEE Sensors J.*, vol. 24, no. 2, pp. 2145–2153, Jan. 2024.

Jinhu Tu received the B.S. degree in software engineering in 2021 from the JiangXi University of Science and Technology, Ganzhou, China, and the M.S. degree in computer science and technology in 2024 from the School of Computer Science and Engineering, Central South University, Changsha, China, where he is currently working toward the doctoral degree in control science and engineering with the School of Control Science and Engineering.

His research interests include safety assessment, digital twin, and self-safety reinforcement learning.

Xiaohong Nian received the B.Sc. degree in mathematics education from Northwest Normal University, Lanzhou, China, in 1985, the M.Sc. degree in fundamental mathematics from Shandong University, Jinan, China, in 1996, and the Ph.D. degree in general mechanics from Peking University, Beijing, China, in 1999.

From 2004 to 2008, he was a Research Fellow with the Institute of Zhuzhou Electric Locomotive, Zhuzhou, China. He is currently a Professor with Central South University, Changsha, China. His current research interests include coordinated control and optimization of complicated multiagent systems, reinforcement learning, and machine learning.

Xunhua Dai received the B.S., M.S., and Ph.D. degrees in control science and engineering from Beihang University, Beijing, China, in 2013, 2016, and 2020, respectively.

Since 2020, he has been an Associate Professor with Central South University, Changsha, China, in computer science and engineering. His main research interests include reliable intelligent control, safety assessment, and design optimization of unmanned aerial robotics.