

Article

Federated Fault Diagnosis for Heterogeneous Satellite Constellations Using Adaptive Dual Knowledge Distillation

Xiaoyu Xing ^{1,2} , Shuyi Wang ^{1,2,*}, Wenjing Liu ^{1,2}, Chengrui Liu ^{1,2} , Hanyu Liang ^{1,2}  and Yan Zhang ^{1,2}

¹ Beijing Institute of Control Engineering, Beijing 100094, China; xingxiaoyu0904@163.com (X.X.); 18010077853@163.com (W.L.); liuchengrui_502@163.com (C.L.); leung_lucy@163.com (H.L.); zhangyan6521@163.com (Y.Z.)

² National Key Laboratory of Space Intelligent, Beijing 100094, China

* Correspondence: wangshuyi78@163.com; Tel.: +86-17863116833

Abstract

Satellite constellation fault diagnosis presents significant challenges, including excessive inter-satellite communication overhead, data heterogeneity, satellite platform heterogeneity, and limited model generalization. To address these issues, this paper proposes an Adaptive Federated Dual Knowledge Distillation (AFDL) framework. First, we develop a heterogeneous federated learning architecture where different satellite types are modeled as independent local clients. Each client incorporates a dedicated local feature extractor to capture its personalized characteristics. Second, a collaborative knowledge distillation mechanism is designed to integrate soft supervision signals from the global server model with personalized feature knowledge from diverse local clients. Finally, a task-specific knowledge distillation strategy is introduced, which employs targeted distillation and dynamic weight adaptation to enhance the generalization performance of local clients. Extensive experiments demonstrate that AFDL achieves 92.3% server accuracy and 84.8% client unseen fault accuracy on our self-collected dataset, outperforming the strongest federated learning baselines by 11.2 and 21.2 percentage points, respectively, while reducing communication overhead by 57.1% compared with centralized learning. Experiments on the public ESA anomaly dataset further confirm the generalization capability of AFDL under real satellite telemetry data. These results validate that AFDL enables efficient and adaptive fault diagnosis across heterogeneous satellite constellations with moderate communication costs.

Keywords: satellite constellation; fault diagnosis; federated learning; knowledge distillation; communication efficiency



Academic Editor: Jose Luis Calvo-Rolle

Received: 8 June 2026

Revised: 3 July 2026

Accepted: 9 July 2026

Published: 11 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

With the advancement of modern aerospace technology, the global demand for space exploration missions has been increasing steadily. The number of spacecraft launches and operational on-orbit vehicles has shown consistent growth compared to previous years. According to incomplete statistics, a total of 263 launch missions were conducted globally in 2024, deploying 2873 spacecraft, primarily consisting of low-earth-orbit constellations such as SpaceX's Starlink (1982 satellites), China's "Qianfan" and "Geely" constellations (over 100 satellites), and Russia's "Citro-AIS" (44 satellites). As of September 2024, the number of commercial satellite constellations in orbit has reached 411 [1]. With the increasingly demanding space science exploration missions, the future development of aerospace

activities will not only involve a continuous increase in launch frequency and on-orbit spacecraft numbers, but also the deployment of diverse scientific instruments onboard. The significant enhancement in system configuration and structural complexity of spacecraft has imposed substantial pressure on ground-based measurement and control operations, thereby urgently necessitating the improvement of safety, reliability, and operational stability of in-orbit constellations. To ensure the successful execution of scientific exploration missions, there are growing requirements for health management technologies applicable to large-scale constellations. Research on constellation fault diagnosis technology holds considerable practical importance for supporting future scientific exploration missions.

In recent years, constellation fault diagnosis technology has garnered increasing attention, currently transitioning from single-satellite independent diagnosis [2] to constellation-cooperative intelligent diagnosis. However, due to the vast scale, high dynamics, and complex operational environment of constellations, fault diagnosis technology still faces multiple challenges. Teng et al. [3] proposed a knowledge graph-based satellite fault diagnosis framework that combines expert rules with data-driven approaches. This effectively addresses sparse data and cold-start problems while enhancing detection and reasoning capabilities for rare faults. Liu et al. [4] constructed a multidimensional system model for low-Earth-orbit constellations. By integrating single-satellite systems, constellation orbital environment systems, and communication link systems, they established a constellation system model covering the physical information space, overcoming the limitations of independent modeling. Han et al. [5] proposed a directed graph model for satellite constellation fault propagation, applying shortest path algorithms to constellation fault diagnosis. This model used directed graphs to characterize fault propagation relationships between satellite nodes, thereby calculating fault propagation probabilities. These approaches consider factors such as the overall constellation state, inter-satellite coupling relationships, and dynamic topology changes. They require real-time transmission of massive data, significantly increasing inter-satellite communication pressure. Furthermore, these approaches rely heavily on prior knowledge and constellation modeling, lacking the ability to learn new fault types in real time. Overall, constellation fault diagnosis faces challenges including insufficient real-time capability, difficulty in predicting fault propagation, and sparse data samples. No mature solution for constellation fault diagnosis has emerged to date, necessitating in-depth research into intelligent methods suitable for this application.

In parallel, intelligent fault diagnosis methods in industrial systems have developed rapidly in recent years. Deep learning-based methods mainly employ neural networks to automatically learn fault representations from monitoring signals and predict fault categories [6–8]. Compared with traditional feature-engineering methods, they can extract nonlinear and high-dimensional fault features more effectively, showing strong diagnostic performance in complex industrial systems. Reinforcement learning-based methods formulate fault diagnosis or maintenance decision-making as a sequential optimization problem, enabling diagnostic models to adaptively select actions or update strategies according to environmental feedback [9,10]. Such methods are particularly useful for dynamic operating conditions and decision-oriented maintenance tasks. More recently, large language model (LLM)-based methods have emerged as a new research hotspot in intelligent fault diagnosis [11,12]. By leveraging their semantic understanding, reasoning, and instruction-following capabilities, LLM-based diagnostic models show potential in fault interpretation, diagnostic report generation, and cross-condition knowledge transfer. However, most existing intelligent fault diagnosis methods are designed for ground-based industrial scenarios with relatively sufficient computational resources, stable communication conditions, and centralized data availability. They are difficult to directly apply to

satellite constellation scenarios, where onboard computing resources are limited, real-time response is required, fault samples are scarce, and raw telemetry data cannot be frequently exchanged among satellites due to inter-satellite communication constraints. In particular, large-scale deep models and LLM-based diagnostic methods usually require high computational and storage costs, which are incompatible with the resource-constrained onboard environment. Therefore, an effective constellation fault diagnosis method should not only improve diagnostic accuracy, but also support distributed learning, reduce communication overhead, and adapt to heterogeneous satellite platforms.

Federated Learning (FL) is a distributed machine learning framework enabling collaborative training of global models among clients without data transmission [13]. Key advantages of FL include robust privacy protection, enhanced model performance, and reduced data transmission overhead. In industrial fault diagnosis, FL integrates multi-source knowledge data to significantly improve model recognition of rare fault patterns while safeguarding data privacy. These strengths have driven its adoption in intelligent industrial fault diagnosis, with numerous studies applying FL to industrial fault diagnosis [14]. Younis et al. [15] addressed the imbalanced distribution of fault data in IoT devices by proposing an improved synthetic minority over-sampling technique. Within the federated learning framework, this method simulated and fused data for rare faults, making it suitable for supervised learning tasks involving Non-Independent and Identically Distributed (Non-Iid) data scenarios. Zhang et al. [16] adapted federated learning methods to the characteristics of mechanical fault diagnosis. By training models using local private data and transmitting only model parameters between clients and servers, they achieved knowledge transfer between models. Kang et al. [17] proposed an unsupervised federated universal domain adaptation method for rolling bearing fault diagnosis, which employs random mapping for privacy protection, mixed radial basis kernel–maximum mean discrepancy for feature alignment, margin vectors for separating unknown classes, and dynamic weighted loss fusion to improve learning efficiency.

The above research demonstrates that federated learning is applicable to industrial fault diagnosis. However, existing methods face significant challenges when directly applied to constellation fault diagnosis, primarily due to three key issues:

- (1) Satellite heterogeneity and data heterogeneity. Constellations typically comprise multiple satellite types with different sensor configurations, system architectures, and mission objectives, resulting in complex failure modes and diverse data formats. Traditional federated learning frameworks, designed for scenarios with homogeneous data and models, struggle to achieve effective knowledge transfer and exhibit poor model generalization capabilities.
- (2) Data transmission constraints under dynamic inter-satellite communications. Operating in complex and harsh space environments, satellite constellations rely on inter-satellite links characterized by limited bandwidth, high latency, and intermittent connectivity. Traditional federated learning requires frequent global model synchronization and data exchange, consuming substantial inter-satellite communication resources and hindering real-time model updates to fault diagnosis.
- (3) Dynamic constellation configuration changes and scarcity of fault samples. Due to real-time variations in constellation operations and environmental conditions, constellation configurations dynamically evolve. The global server cannot maintain real-time connections with all clients, necessitating local client models capable of diagnosing all faults across identical satellite models. However, individual satellites experience few faults during operation, leading to insufficient training samples. This results in local models overfitting and poor generalization, hindering accurate diagnosis of unknown faults.

To address these limitations collectively, this paper proposes a novel Adaptive Federated Dual Knowledge Distillation (AFDL) framework. Its core innovations are threefold:

- (1) To address model and data heterogeneity, a multimodal heterogeneous federated architecture and a feature alignment mechanism are designed. Multimodal local client models are established based on different satellite models. Local feature extractors capture personalized client features, overcoming the rigidity of unified modeling in traditional federated learning. A local feature alignment mechanism is proposed to aggregate multiple clients' local features at the global server, enabling knowledge transfer across satellite models.
- (2) To address excessive inter-satellite communication pressure, a lightweight bidirectional knowledge distillation strategy is proposed. Clients perform local knowledge distillation via sparse feature aggregation, transmitting local feature vectors to the server. The server applies soft-target distillation to clients, using the server model's diagnosed class probability distribution as a supervision signal. This approach transmits only lightweight log probabilities instead of raw data to clients, significantly reducing communication volume per round.
- (3) To address dynamic constellation topology and scarce fault samples, we propose a two-stage dynamic knowledge distillation strategy. The collaborative bidirectional distillation stage optimizes the server's global diagnostic capability for known faults, and the subsequent task-specific stage enhances local models' generalization to unseen faults using a frozen server teacher. Trained models can autonomously diagnose all faults across identical satellites without real-time communication.

The subsequent sections of this paper are structured as follows: Section 2 reviews related work; Section 3 details the AFDL model framework and computational workflows for each component; Section 4 presents simulated constellation fault data and experimentally validates the AFDL model's effectiveness; Section 5 summarizes the research findings and proposes future research directions and development recommendations.

2. Related Work

2.1. Constellation Fault Diagnosis

Research on constellation fault diagnosis is currently in its infancy, with limited relevant studies available for reference. Yan et al. [18] proposed a Global Navigation Satellite System (GNSS) multi-fault isolation method based on an extended greedy search. They introduced an incremental measurement set expansion strategy to construct a minimal basic set and rank residuals accordingly. This approach reduces the "drowning effect" caused by false deletion of valid measurements, enabling rapid and accurate multi-fault detection and isolation in constellation GNSS positioning systems. MENG et al. [19] designed a non-least-squares estimator that moderately increases nominal positioning errors to mitigate integrity risks. They proposed an advanced autonomous receiver integrity monitoring method that eliminates constellation-to-constellation differences through constellation-weighted averaging. Huang et al. [20] proposed a Bayesian approach for multi-constellation navigation, computing Bayesian posterior probability boundaries that incorporate multiple concurrent fault assumptions. This enhances the robustness of integrity risk assessment without relying on optimistic prior fault distributions. They establish Bayesian fault-tolerant protection levels to constrain position estimation errors and evaluate integrity availability.

2.2. Federated Learning

Federated learning is a distributed machine learning framework that enables clients to train models locally while periodically exchanging data knowledge with a central server.

Based on data distribution characteristics, federated learning is categorized into Horizontal Federated Learning (HFL) [21], Vertical Federated Learning (VFL) [22], and Federated Transfer Learning (FTL) [23]. Horizontal Federated Learning primarily learns based on samples, which is suitable when local clients share identical dataset features but differ in sample types. Vertical Federated Learning focuses on feature-based learning, applicable when sample types are identical but dataset features vary. Federated Transfer Learning integrates both sample and feature learning, addressing scenarios with different dataset features and sample types.

Satellite constellations comprise diverse satellite models and architectures, which leads to significant differences in both dataset characteristics and sample types. Federated Transfer Learning is suitable for this scenario. Therefore, this section primarily reviews FTL research. Jiang et al. [24] proposed ObsBattery, a federated learning framework for satellite battery prediction that addresses the challenge of data heterogeneity caused by varying orbital sunlight exposure through position-aware clustering. Zhang et al. [25] addressed the challenge of efficiently managing massive heterogeneous time-series data in industrial IoT (IIoT) by applying FL to device data management in wireless network environments. They proposed a deep reinforcement learning FL algorithm that enhances model aggregation efficiency and communication economy. Lu et al. [26] proposed a Compound Imbalanced Privacy-Preserving Federated Learning (CI-PPFL) framework for distributed wind turbine fault diagnosis. They introduced a parameter proportional update strategy and a gradient self-monitoring mechanism to accurately identify global imbalance information. Lin et al. [27] introduced Federated Distillation Fusion (FedDF), a model fusion method based on ensemble distillation. It trains a central classifier using unlabeled data generated from client model outputs, enabling flexible aggregation of heterogeneous client models. Wu et al. [28] proposed FedKD, a communication-efficient federated learning method, which conducts adaptive mutual knowledge distillation between large local mentor models and shared small mentee models on each client, and further compresses transmitted gradients dynamically via SVD to sharply reduce communication overhead while maintaining competitive model performance. Wu et al. [29] proposed Global Prototype Distillation (FedGPD) for heterogeneous federated learning, which aggregates class-level feature prototypes from all clients to form global prototypes, and takes such global prototype knowledge as distillation signals to regularize local client training, so as to alleviate local model drift caused by non-IID data without relying on public proxy datasets. This approach mitigates feature distribution skew and resolves data feature heterogeneity issues.

2.3. Research Gaps

In summary, existing research has laid a solid foundation for constellation fault diagnosis and federated learning. However, three critical research gaps remain:

- (1) Lack of a heterogeneity-aware FL framework that can simultaneously handle both satellite platform heterogeneity and data heterogeneity.
- (2) Insufficient attention to communication efficiency in FL-based fault diagnosis methods, making them impractical for resource-constrained satellite systems.
- (3) Limited research on enhancing local model generalization to unseen faults, which is essential for autonomous constellation operations.

This paper addresses these gaps by proposing the AFDL framework, which combines heterogeneous federated architecture design, lightweight bidirectional distillation, and two-stage adaptive training to achieve efficient and robust fault diagnosis in heterogeneous satellite constellations. To empirically validate the superiority of our method, we select three representative heterogeneous federated learning baselines for comparison, including FedDF, FedKD, and FedGPD. The core methodological differences are detailed in Table 1.

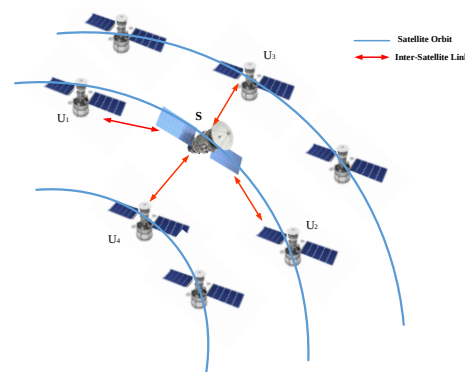
Table 1. Core comparison between AFDL and heterogeneous federated learning baselines.

Method	Heterogeneity Support	Distillation Mechanism	Unseen Fault Generalization
FedDF	Only model architecture heterogeneity	One-way output-level distillation; requires proxy data	None
FedKD	Only model heterogeneity	Local mentor–mentee mutual distillation	None
FedGPD	Only data distribution heterogeneity (non-IID)	One-way feature-level prototype distillation	None
AFDL	Platform + data dual heterogeneity	Bidirectional feature & output distillation, lightweight	Two-stage adaptive dynamic weighting

3. Adaptive Federated Dual Knowledge Distillation Framework

3.1. Constellation Heterogeneous Federated Learning Model Architecture

The heterogeneous federated learning framework comprises a central server and multiple local clients. Clients utilize local data to train local models and upload distilled knowledge to the server. The central server aggregates data knowledge from local clients to train and update the global model, while guiding the training and learning of local client models. Satellites within the constellation can be categorized by computational capability into data relay satellites and conventional satellites. Data relay satellites possess high-performance computing units with computational power far exceeding that of conventional satellites. They serve as regional computing centers for the constellation, functioning as servers. Conventional satellites, with weaker computational capabilities, act as local clients, processing only local data. Based on their structure and type, different neural network model architectures are configured for conventional satellites. This paper classifies them into large clients and small clients according to structural complexity. Typically, data relay satellites are equipped with four laser terminals mounted on opposite sides of the satellite platform (two pairs installed in opposite directions) to enable forward, backward, left, and right connections along the primary orbital direction. At any given moment, the forward and rear ports of a data relay satellite S connect to co-orbital satellites U_1 and U_2 , while the forward and rear ports connect to cross-orbital satellites U_3 and U_4 , as shown in Figure 1. During operation, the data relay satellite maintains constant connections with adjacent satellites in the same orbital plane, while connections with satellites in different orbital planes dynamically change. Thus, over its orbital period T , it communicates via inter-satellite links with n conventional satellites. Taking one orbital period of S as an example, its heterogeneous federated learning architecture is shown in Figure 2.

**Figure 1.** Inter-satellite link structure.

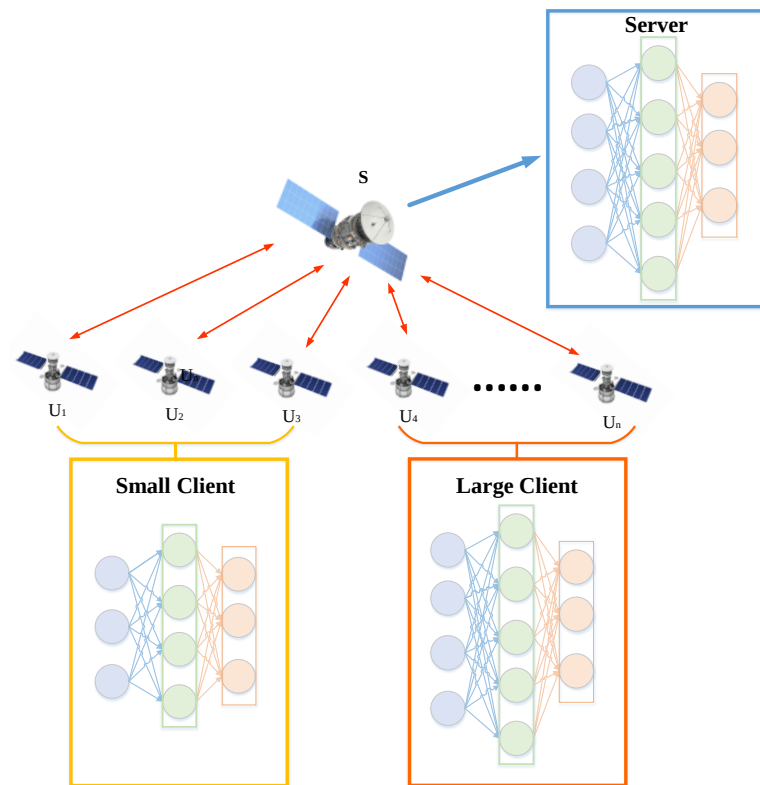


Figure 2. Heterogeneous constellation federated learning model architecture.

3.2. Algorithm Workflow

The AFDL algorithm employs a two-stage collaborative distillation framework. It jointly optimizes the local client's diagnostic capability for fault data of the same model and the server's global diagnostic capability.

1. Collaborative Bidirectional Distillation Phase. The optimization objectives are to enhance the server's global diagnostic effectiveness and improve clients' learning of known fault feature knowledge. Each client trains models based on locally seen data and extracts personalized features. Their output distributions and personalized features are uploaded to the server for knowledge alignment, enabling server-side aggregation updates. The updated server then downloads soft labels to clients, minimizing the discrepancy between client predictions and server soft labels using KL divergence. This feature knowledge transfer between clients and servers is termed bidirectional knowledge distillation.
2. Task-specific Distillation Phase. When the server achieves the required global diagnostic rate, the task-specific distillation phase commences. The optimization goal is to enhance the client's generalization capability for unseen faults. The server parameters are frozen as a fixed teacher to guide the client's unknown data training. Simultaneously, the client trains based on locally known data to maintain its diagnostic capabilities for known faults. The client adjusts its training focus through a dynamic weighting mechanism, concentrating on enhancing diagnostic capabilities for unknown fault categories.

The core difference between the two phases is that Phase 1 focuses on global knowledge fusion and server performance improvement via bidirectional feature and output alignment, while Phase 2 focuses on local generalization enhancement via fixed teacher distillation and dynamic weight adaptation. The overall workflow is illustrated in Figure 3, and detailed implementations of each phase are presented in the following subsections.

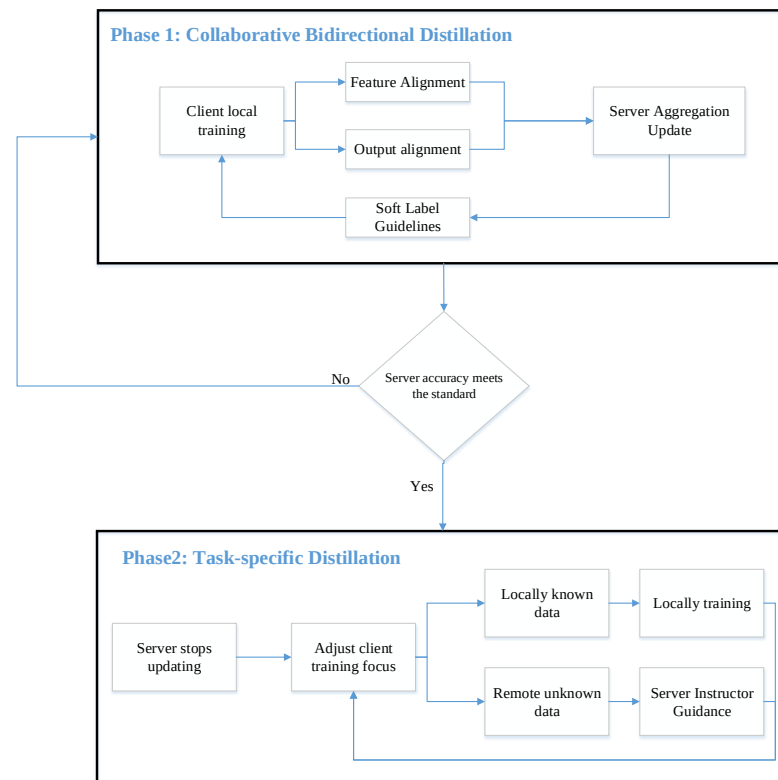


Figure 3. AFDL algorithm workflow.

3.3. Collaborative Bidirectional Distillation Phase

Due to the varying types of satellites within a constellation and their differing component configurations, the fault data structures exhibit significant difference. Consequently, specific neural network architectures must be tailored for different satellite types. This paper assumes the constellation comprises two satellite categories: one with fewer sensors and simpler local client neural networks, termed small clients, and another with more sensors and more complex local client neural networks, termed large clients. To ensure the server possesses the capability to process the most complex data, its network architecture is configured identically to that of the large client. The collaborative bidirectional distillation phase comprises the following three steps: (a) Clients train models and extract personalized features based on local data, then upload knowledge to the server; (b) Server knowledge distillation aggregates personalized features and classification predictions from all clients to update server model parameters; (c) Clients perform reverse knowledge distillation, receiving server guidance to optimize local models based on global soft labels and global features. The workflow is illustrated in Figure 4.

This bidirectional distillation framework is designed with two explicit optimization objectives. For the client-to-server upward distillation, the core target is to fuse heterogeneous fault knowledge across different satellite platforms into a unified global model, bridging the feature distribution gap caused by inconsistent sensor configurations and network architectures. For the server-to-client downward distillation, the core target is to deliver globally consolidated diagnostic knowledge back to each local satellite, mitigating local model drift and overfitting caused by skewed, limited local fault datasets. Unlike conventional one-way federated distillation schemes that only optimize the global model, this bidirectional interactive design achieves joint improvement of global fusion accuracy and local generalization capability, which better matches the distributed autonomous operation requirements of satellite constellations.

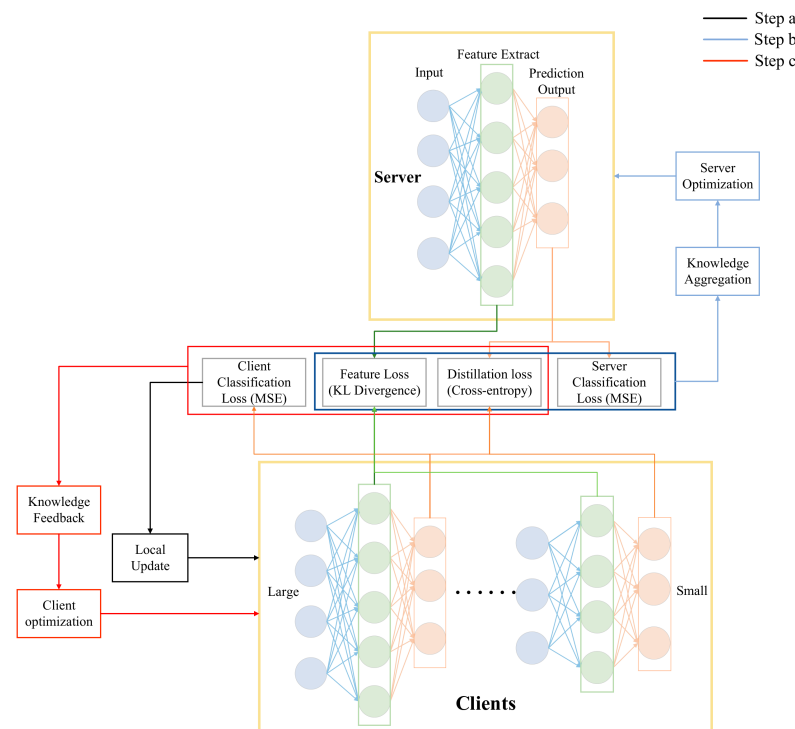


Figure 4. Collaborative bidirectional distillation workflow.

In terms of knowledge transfer hierarchy, the framework implements dual-level distillation: output-level distillation based on class probability distributions and feature-level distillation based on latent feature vectors. The feature alignment module acts as the core carrier of feature-level knowledge transfer: it maps the personalized features extracted by heterogeneous client networks into a unified latent space through a linear adaptation layer, eliminating feature distribution gaps caused by inconsistent sensor dimensions and model architectures, and laying the foundation for effective fusion of multi-source heterogeneous fault knowledge.

3.3.1. Client Local Training

The client-side local training phase primarily trains client models using local data to enhance the client's known fault diagnosis capabilities. Each client comprises an input layer, hidden layers, and an output layer. This paper designates specific hidden layers as feature extraction layers to extract personalized features from local data for subsequent knowledge distillation. Since data across different clients remains mutually invisible, the local dataset $D_{ks} = \{x_i, y_i\}_{i=1}^n$ of client k is treated as locally known faults, termed seen data, where x represents the true data values and y denotes the true data labels. Data from other clients is treated as an unknown fault, termed unseen data. The output h_1 of hidden layer 1 is obtained by multiplying input data X_i by the weights W_1 of the first hidden layer and adding the bias term b_1 :

$$h_1 = f_1(W_1 X_i + b_1) \quad (1)$$

$f_1(\cdot)$ is the activation function, with the sigmoid function selected for its ease of forward propagation. The output of hidden layer 1 serves as the input for the next hidden layer 2. By repeating the above equation and calculating up to the output of the feature extraction layer, the latent feature vector h_{client}^i can be obtained. Computing the final hidden layer vector h_t with weights W_y and bias b_y produces the output prediction classification vector p_{client}^i :

$$p_{client}^i = f_o(W_y h_t + b_y) \quad (2)$$

$f_o(\cdot)$ is the activation function, with the *softmax* function selected for multi-class classification, outputting the predicted probabilities for each class. The cross-entropy function is used to compute the classification loss L_{cls}^{client} for client k :

$$L_{cls}^{client} = -\frac{1}{n} \sum_{i=1}^n \sum_{c \in C_{seen}} y_{i,c} \log(p_{i,c}) \quad (3)$$

C_{seen} denotes the set of visible categories for client k , $p_{i,c}$ represents the predicted probability of category c by p_{client}^i , and $y_{i,c}$ indicates the true label of sample i in category c . Based on this, gradient descent is employed to optimize the parameters of the client's neural network. Cross-entropy naturally matches the *softmax* output layer for multi-class fault diagnosis. It avoids gradient saturation on sparse local fault samples and directly measures prediction-label distribution mismatch, providing a stable optimization baseline for local known fault recognition.

3.3.2. Server Knowledge Distillation

Within the collaborative bidirectional distillation framework, the server computes optimized classification loss, feature alignment loss, and distillation loss to achieve multi-level global knowledge distillation learning and enhance model generalization capabilities. Classification loss defines task objectives, feature alignment loss constructs a unified knowledge representation space, and distillation loss extracts client-side knowledge obtained through distillation. The resulting server global model exhibits robust fault diagnosis capabilities while adapting to the heterogeneous data characteristics of diverse clients.

After completing local training on the client side, the server aggregates the latent features and prediction probability distributions from all N clients to generate the global feature center h_{global} and the average soft label p_{global} :

$$h_{global} = \frac{1}{N} \sum_{i=1}^N h_{client}^i \quad (4)$$

$$p_{global} = \frac{1}{N} \sum_{i=1}^N p_{client}^i \quad (5)$$

The classification loss L directly optimizes the server model's ability to distinguish true labels. This ensures accurate fault classification while avoiding overreliance on client-side knowledge. It is computed using the cross-entropy function:

$$L_{cls}^{server} = -\sum y_{true} \log p_{server} \quad (6)$$

y_{true} represents the true label of the server data, and p_{server} represents the server's predicted classification vector. Cross-entropy maintains objective consistency with client-side training. Hard label supervision anchors the server's training direction, preventing model drift from ambiguous soft distillation signals, which is critical for high-reliability aerospace scenarios.

Feature Alignment Loss L_{feat}^{server} is the core component of feature-level knowledge distillation, which forces the server feature space to align with the aggregated global feature center from all clients, fundamentally addressing feature distribution shifts caused by satellite platform heterogeneity, sensor configuration differences, and non-IID data distribution. Heterogeneous clients first extract personalized latent features through their dedicated feature extractors, and then map features of different original dimensions to a unified latent space through a trainable linear feature adapter layer. The server aggregates the adapted features of all participating clients to obtain a global feature center h_{global} ,

and constrains the feature output of the server model to approach this global center through alignment loss, so that fault features of the same category from different satellite platforms can be clustered in the unified latent space. This enables the global model to uniformly interpret heterogeneous feature distributions across all clients, and effectively avoids the problem of invalid knowledge transfer caused by feature space misalignment in cross-platform distillation.

This paper selects the L_2 norm (Mean Squared Error, MSE) as the feature alignment loss function, which is calculated as:

$$L_{feat}^{server} = \|h_{server} - h_{global}\|_2 = \sqrt{(h_{server} - h_{global})^2} \quad (7)$$

$\|\cdot\|_2$ denotes the L_2 norm, and h_{server} represents the latent feature vector of server data. The L_2 norm directly measures the Euclidean distance of continuous latent features with low computational overhead, adapting to resource-constrained on-board devices. It provides smooth gradients for stable federated iteration and fits the unified feature space design without extra conversion steps.

Distillation loss $L_{distill}^{server}$ represents the discrepancy between server predictions and client distilled knowledge. It enhances the server's ability to learn edge knowledge and improves fault diagnosis capabilities for rare failures. It is calculated using KL divergence:

$$L_{distill}^{server} = D_{KL}(p_{global} \| p_{server}) = \sum_{i=1}^C p_{global}^i \log\left(\frac{p_{global}^i}{p_{server}^i}\right) \quad (8)$$

C denotes the total number of data categories, and $D_{KL}(\cdot)$ represents the KL divergence function. KL divergence specializes in measuring probability distribution differences, fully retaining inter-fault correlation information that hard labels cannot provide. It aligns with the core goal of knowledge distillation and maintains stable gradients under temperature scaling.

Weighted summation of the above three loss categories yields the total server loss, thereby optimizing the server's global diagnostic capabilities:

$$L_{server} = \lambda_1 L_{cls}^{server} + \lambda_2 L_{feat}^{server} + \lambda_3 L_{distill}^{server} \quad (9)$$

λ represents the loss weighting coefficient, which is manually set based on the experimental objectives and focus areas.

3.3.3. Client Reverse Knowledge Distillation

In the collaborative bidirectional distillation framework, client-side reverse distillation refers to the process of knowledge flowing back from the server to the client. The client receives globally distributed knowledge from the server and locally computes classification loss, feature alignment loss, and distillation loss. This guides the client model to further optimize local feature representations and classification predictions, reducing feature distribution discrepancies caused by data heterogeneity. It prevents the client from overfitting due to limited local data, thereby enhancing the client's generalization capability on unseen data.

The calculation principles for the client's classification loss L_{cls}^{client} , feature alignment loss L_{feat}^{client} , and distillation loss $L_{distill}^{client}$ are consistent with those on the server, though the knowledge flow direction is reversed.

Different from the server-side alignment that pursues global feature fusion, the client-side feature alignment aims to guide the local feature space to evolve toward the globally unified feature space calibrated by the server, so as to correct the local model drift

caused by limited and skewed local data. The client takes the server-issued global standard feature vector as the alignment target, and uses the same $L2$ norm loss to constrain its own feature output. This can not only retain the personalized feature extraction capability of the local model for its own satellite platform data, but also make its feature distribution compatible with the global knowledge system, laying a foundation for the subsequent generalization of unseen fault categories in the task-specific distillation phase.

Specifically, cross-entropy is adopted for classification loss to anchor the diagnostic accuracy of visible faults and prevent performance degradation under distillation guidance. The $L2$ norm is retained for feature alignment loss, with low computational overhead adapting to on-board resource constraints. KL divergence is used for distillation loss to absorb global inter-class correlation knowledge and alleviate local overfitting caused by sparse and skewed samples.

Accordingly, we obtain:

$$L_{cls}^{client} = -\sum y_{true} \log p_{client} \quad (10)$$

$$L_{feat}^{client} = \|h_{client} - h_{server}\|_2 = \sqrt{(h_{client} - h_{server})^2} \quad (11)$$

$$L_{distill}^{client} = D_{KL}(p_{server} \| p_{client}) = \sum_{i=1}^C p_{server}^i \log \left(\frac{p_{server}^i}{p_{client}^i} \right) \quad (12)$$

The client enhances the local diagnostic capability and generalization capability of the total loss model, where the total loss L_{client} is defined as:

$$L_{client} = \lambda_1 L_{cls}^{client} + \lambda_2 L_{feat}^{client} + \lambda_3 L_{distill}^{client} \quad (13)$$

3.4. Task-Specific Distillation Phase

During the task-specific distillation phase, dynamic adjustments to training weights and knowledge transfer strategies significantly enhance the client model's recognition capabilities for unseen data categories. This phase freezes the server model parameters and halts server training. The server acts as a teacher, guiding the client to learn and predict specifically for unseen data; simultaneously, the client trains locally on seen data to maintain recognition capabilities for known data.

First, freeze the server model parameters and halt server optimization. The frozen server transmits teacher guidance, including soft predictions and global feature representations, to guide client-side unseen-category learning. Calculate the visible loss L_{seen} for the client using formula (3) and the unseen loss L_{unseen} using formula (13). The total loss for client k adopts a linear weighted combination form with dynamic adjustment coefficients:

$$L_k^{total} = \alpha_t L_{unseen} + \beta_t L_{seen} \quad (14)$$

where α_t and β_t denote the loss weights for unseen and seen categories at training round t , respectively. The weights are updated per round adaptively according to the client's local unseen fault diagnosis performance, following the rules below:

$$\alpha_t = \text{Clip}(\alpha_{t-1} + \eta \cdot \Delta acc_{unseen}, \alpha_{\min}, \alpha_{\max}) \quad (15)$$

$$\beta_t = 1 - \alpha_t \quad (16)$$

Δacc_{unseen} is the accuracy difference in unseen categories between the current round t and the previous round $t - 1$, calculated on the client's local validation set. η is the

fixed weight update step size, and the $\text{Clip}(\cdot)$ function constrains α_t within the interval $[\alpha_{\min}, \alpha_{\max}]$ to avoid extreme weight imbalance.

When $\Delta acc_{unseen} > 0$, unseen fault learning progresses effectively, and α_t increases moderately to strengthen the guidance of global teacher knowledge and accelerate convergence of unseen generalization. When $\Delta acc_{unseen} \leq 0$, learning stagnates or degrades, and α_t remains unchanged to prevent overfitting to unseen data while stabilizing seen fault performance. The entire weight update process is executed locally on each client without extra communication with the server, which automatically adapts to heterogeneous learning progress of different satellite platforms and eliminates the need for per-client manual tuning. The initial values of α and β are set empirically according to the sample ratio of seen to unseen fault categories.

For clarity and reproducibility, the complete two-stage training pipeline of the AFDL framework, covering bidirectional distillation, phase switching, and dynamic weight adaptation, is formalized in Algorithm 1.

Algorithm 1. Complete Two-Stage Training Procedure of AFDL

Input:

Private datasets D_{ks} for $k = 1, 2, \dots, N$;
 Initial global server model $f_{\theta_{server}}$;
 Initial local client models f_{θ_k} ;
 Phase transition thresholds $acc_{server}^{thr}, acc_{seen}^{thr}$;
 Initial loss weights for Phase 2: α_0, β_0 ;
 Weight update step size η , weight bounds $[\alpha_{\min}, \alpha_{\max}]$;
 Maximum communication rounds T_1 (Phase1), T_2 (Phase1);
 Number of selected clients per round K ;

Output:

Trained global server model $f_{\theta_{server}}$;
 Trained local client models f_{θ_k} ;

I Phase 1: Collaborative Bidirectional Distillation

For round $t = 1, 2, \dots, T_1$ do:

1. Client Local Training Process

For each client $k = 1, 2, \dots, N$ in parallel do

For each batch $\{x_i, y_i\}_{i=1}^n \in D_{ks}$ do

- ① Extract local features and predictions h_{client}^i and p_{client}^i
- ② Compute classification Loss L_{cls}^{client}
- ③ Update client models f_{θ_k}
- ④ Upload to server: Local features h_{client}^i and predictions p_{client}^i

2. Server Knowledge Distillation Process

For server do

- ① Aggregate uploaded knowledge:
Global feature center h_{global} ; Average soft labels p_{global}
 - ② Compute total server loss L_{server} :
Server classification loss L_{cls}^{server} ; Distillation loss $L_{distill}^{server}$; Feature alignment loss L_{feat}^{server}
 - ③ Update server model $f_{\theta_{server}}$
-

Algorithm 1. Cont.**3. Client Reverse Distillation****For** each client $k = 1, 2, \dots, N$ **do**① Download from server: Updated global soft labels p_{server} ; Server feature Vector h_{server}

② Compute total client loss:

Client classification loss L_{cls}^{client} ; Distillationloss L_{feat}^{client} ; Feature alignment loss L_{feat}^{client} ③ Update client models f_{θ_k} **4. Check phase transition condition**If server accuracy $\geq acc_{server}^{thr}$ and all clients' seen accuracy $\geq acc_{seen}^{thr}$:

Break and proceed to Phase 2.

II: Phase 2: Task-specific Distillation1. Freeze server model parameters θ_{server} .2. Server distributes teacher guidance to each client k : server latent features h_{server} , server soft predictions p_{server} .3. Initialize: $\alpha \leftarrow \alpha_0, \beta \leftarrow \beta_0, acc_{unseen}^{(0)} \leftarrow 0$.**For** round $t = 1, 2, \dots, T_2$ **do**:**For** each client $k = 1, 2, \dots, N$ **do**:① Compute seen loss L_{seen} via cross-entropy classification loss.② Compute unseen loss L_{unseen} .③ Evaluate current unseen accuracy $acc_{unseen}^{(t)}$ on local validation set.③ Calculate accuracy improvement: $\Delta acc_{unseen} = acc_{unseen}^{(t)} - acc_{unseen}^{(t-1)}$ ② Update loss weights α_i dynamically.② Compute total training loss L_{total}^k ② Update local model parameters θ_k via gradient descent.**4. Experiment**

This section evaluates the proposed AFDL framework on two datasets: a proprietary in-orbit satellite control system dataset and the public ESA anomaly dataset. To improve readability, each dataset evaluation is organized into three parts: dataset and parameter settings, main experimental results, and comparative analysis including ablation and baseline comparisons. The common evaluation protocol, including repeated random trials, mean $\pm$ standard deviation reporting, and communication-overhead measurement, is shared by both datasets unless otherwise specified.

4.1. Proprietary In-Orbit Satellite Control System Dataset Experiment**4.1.1. Experimental Data**

To replicate the model heterogeneity and data heterogeneity inherent in constellation operations, this study selected fault data from two in-orbit satellite control systems. Both systems are equipped with varying numbers of star sensors, gyroscopes, and momentum wheels. Satellites with fewer components serve as small clients (16 units), processing 26-dimensional local data; those with more components function as large clients (16 units), processing 33-dimensional local data. Each fault sample spans 900 min with a sampling frequency of 0.5 s. Ten fault types were selected for each client category, totaling 20 fault types. Each local client stores 18,000 data points: 12,000 normal data points and 6000 fault data points. The data is divided into training and test sets at a 7:3 ratio. A portion of the data is used for dimensionality reduction visualization, as shown in Figure 5.

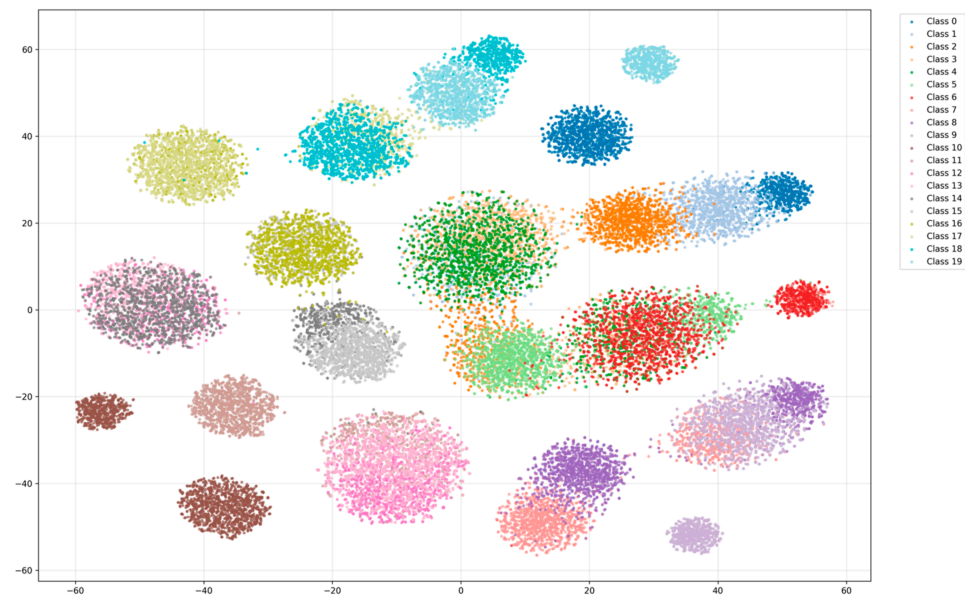


Figure 5. Experimental data visualization.

To simulate data isolation among different clients, this paper configures each client to contain only 3–4 types of fault data locally, with data from different client types remaining mutually isolated. The data distribution among clients is shown in Figure 6, where clients 0–15 are small clients and clients 16–31 are large clients. Data types 0–9 represent local data for small clients, while types 10–19 represent local data for large clients.

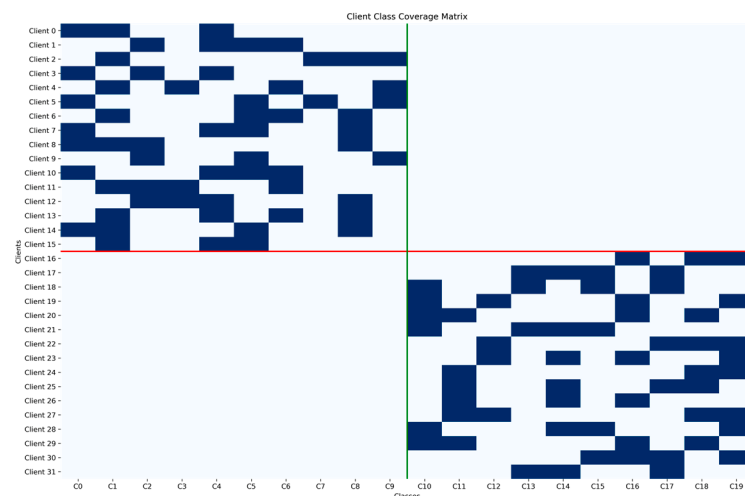


Figure 6. Client data distribution chart.

4.1.2. Parameter Settings

The experimental hardware environment consisted of an AMD Ryzen 7 5800H CPU, an NVIDIA GeForce RTX 3060 GPU, and 16 GB of RAM. The software environment was based on Python 3.10. Neural networks with different architectures were configured for small clients, large clients, and the server. Since the server was required to achieve global diagnostic capability while keeping computational costs low, its architecture was set to be identical to that of the large client. For the low-dimensional data from small clients, zero-padding was applied in advance to expand the input dimension to 33 for unified processing. The neural network architecture is shown in Table 2, and the training parameters are listed in Table 3.

Table 2. Neural network architecture configuration.

	Small Client	Large Client	Server
Input Dimension	26	33	33
Feature Extractor	Linear	Linear	Linear
	(26, 256)	(33, 512)	(33, 512)
	Linear	Linear	Linear
	(256, 128)	(512, 128)	(512, 128)
Feature Alignment	Linear	Linear	Linear
	(128, 128)	(128, 128)	(128, 128)
Classifier	Linear	Linear	Linear
	(128, 20)	(128, 20)	(128, 20)
Dropout	0.3	0.3	0.3
Feature Dimension	128	128	128
Output Category	20	20	20
Total parameter count	92 K	145 K	145 K

Table 3. Training process parameter settings.

Parameters	Phase 1	Phase 2	Instruction
Learning Round	60	60	Early Stopping with Conditions
Client selection per round	10/32	10/32	Random Uniform Sampling
Local training epoch	2	2	Client Local Update
Batch size	64	64	Data Loading Batch
Client learning rate	0.01	0.01	Adam Optimizer
Server learning rate	0.001	0.001	Adam Optimizer
Feature Alignment Weight	0.3	0.3	MSE Loss Coefficient
Distillation Loss Weight	0.5	Dynamic Adjustment	KL Divergence Loss Coefficient
Temperature Parameters τ	2.0	2.0	Knowledge Distillation Temperature
Weight Adjustment	Fixed	Dynamic Adjustment	Improvement on Unseen Classes

The training process was divided into two phases, corresponding to collaborative bidirectional distillation and task-specific distillation, respectively. In this experiment, the phase transition condition was defined as server accuracy > 85% and local seen-data accuracy > 95% for all clients, or when the maximum number of training rounds was reached. The second training phase was terminated when the average local unseen-data accuracy of clients exceeded 90% while the local seen-data accuracy remained above 95%, or when the maximum number of training rounds was reached. To reduce the influence of random initialization and client sampling fluctuations, all comparative experiments and ablation studies were repeated 10 times with distinct random seeds, which controlled network initialization, data shuffling, and client selection. All results related to diagnostic accuracy, training time, experimental communication latency, and communication overhead are reported as mean $\pm$ standard deviation.

4.1.3. Experimental Results

The main result on the proprietary dataset is that AFDL achieves high server-side diagnostic accuracy while substantially improving client-side unseen-fault recognition with moderate communication overhead. Across the 10 independent runs, the proposed method completed the first training phase after 30 ± 8 communication rounds and completed the second training phase after 60 communication rounds. The final average seen-data accuracy on clients reached $99.58\% \pm 0.32\%$, while the unseen-data accuracy was $84.86\% \pm 2.62\%$. The server-side diagnostic accuracy reached $92.36\% \pm 1.45\%$, with an

F1 score of $89.40\% \pm 2.10\%$ and a recall rate of $77.75\% \pm 3.45\%$. In terms of communication overhead, the total communication volume during training was 412.64 ± 42.36 MB, and the average experimental communication latency was 1.28 ± 0.18 s per round. The total training duration was 69.73 ± 9.84 s, and the average client upload/download traffic was 5.92 ± 0.67 MB.

Among the 10 repeated experiments, the best-performing run was selected to visualize the actual diagnostic effect. These conditions were met in the 27th round, which triggered the phase transition, and the second phase stopped at a maximum number of training rounds of 60. The final average seen accuracy on clients was 99.83%, with unseen accuracy at 87.48%. Server accuracy reached 93.7%, achieving an F1 score of 91.50% and a recall rate of 81.20%. The total communication volume during training was 379.19 MB, with an average experimental per-round training and communication time of 1.16 s. The total training duration was 69.73 s, and the average client upload/download traffic was 5.92 MB. It should be noted that this study evaluates communication efficiency mainly from the perspective of transmitted data volume. Physical inter-satellite link effects, such as propagation delay, link acquisition delay, dynamic routing, and queueing delay, are not explicitly modeled and will be considered in future work.

The training curve is shown in Figure 7, and the server confusion matrix is presented in Figure 8. As shown in Figure 7, the server accuracy increases rapidly during the collaborative bidirectional distillation phase and becomes stable after the phase transition, indicating that the first stage effectively improves global knowledge aggregation. Meanwhile, the client-side unseen accuracy continues to increase during the task-specific distillation phase, while the seen accuracy remains at a high level, demonstrating that the second stage improves unseen-fault generalization without significantly sacrificing seen-fault recognition. As reflected by the per-class results in the server confusion matrix in Figure 8 and the client-side unseen evaluation, AFDL achieves stable recognition across most fault categories. This indicates that the improvement in unseen accuracy is broadly consistent rather than dominated by a few easily recognized classes.

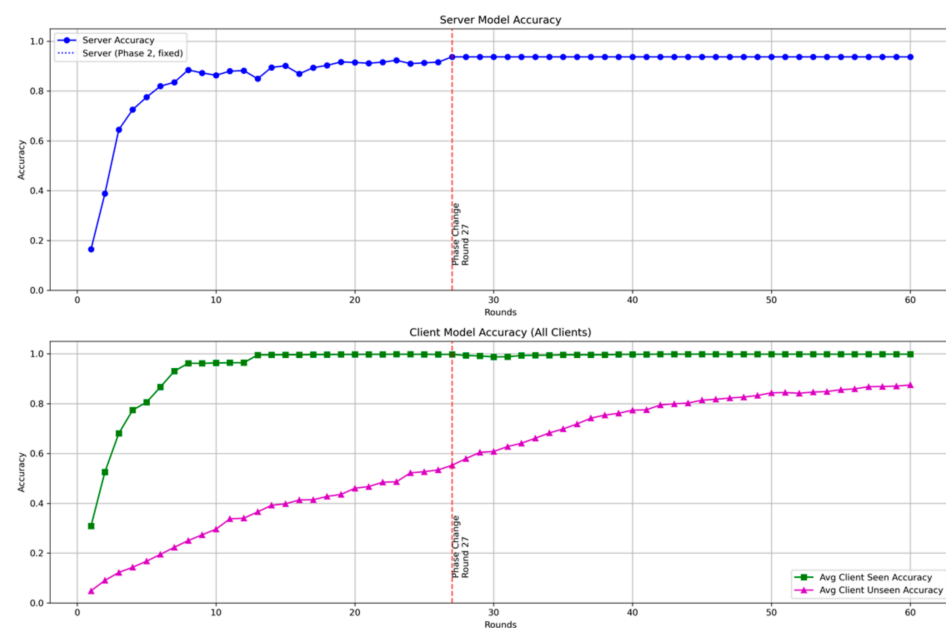


Figure 7. Training process curve.

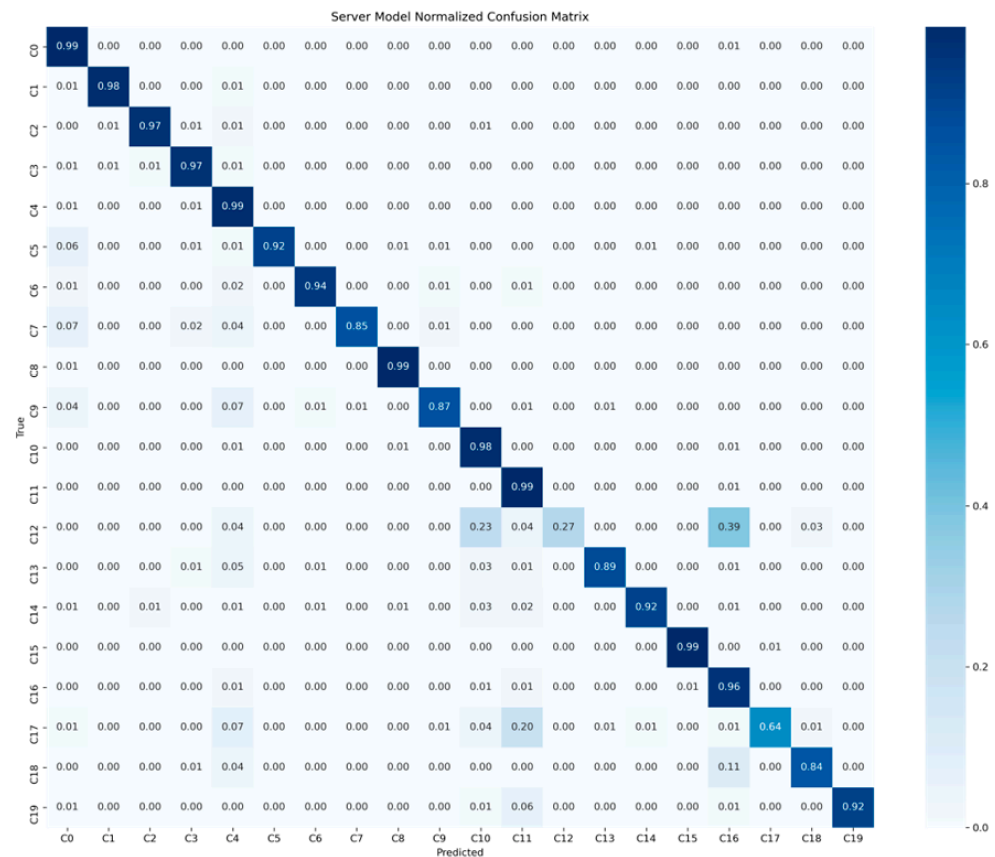


Figure 8. Server confusion matrix.

4.1.4. Ablation Experiments

To verify the performance improvements brought by the proposed feature alignment mechanism, lightweight bidirectional knowledge distillation and task-specific distillation phase for fault diagnosis under heterogeneous data and heterogeneous model scenarios, three ablation variants are designed in this section.

The first variant, denoted as M1, is constructed to quantify the contribution of the feature alignment mechanism to heterogeneous knowledge transfer. Specifically, the trainable feature adapter layer is removed from both client and server architectures, and the feature alignment loss term is excluded from the total loss function of the collaborative bidirectional distillation phase.

The second variant, denoted as M2, is used to verify the communication efficiency gain from lightweight bidirectional knowledge distillation. Instead of transmitting compact latent feature vectors and soft probability distributions between clients and the server, M2 adopts the traditional federated parameter uploading scheme that transfers complete network weights in both directions, removing the lightweight bidirectional knowledge transmission pipeline.

The third variant, denoted as M3, is designed to validate the enhancement effect of the task-specific distillation phase on the generalization performance for unseen faults. In this variant, the task-specific distillation phase is removed, and the model outputs the results directly upon the completion of the first training phase.

For all three variants, all model configurations and dataset settings remain completely consistent with the full AFDL framework except for the ablated modules, to ensure strict single-variable control. Specifically, M1, M2, M3, and AFDL use the same client partition, seen/unseen fault split, client sampling sequence, local training epochs, batch size, optimizer settings, model-capacity configuration, and maximum communication-round budget.

Therefore, the differences reported in Table 4 can be attributed to the removed modules rather than differences in data partitioning, training budget, or model scale.

Table 4. Comparison of variant methods.

Indicator	M1	M2	M3	AFDL
Server Accuracy Rate	67.8% $\pm$ 3.1%	90.1% $\pm$ 3.6%	91.9% $\pm$ 2.0%	92.3% $\pm$ 1.4%
Mean Client Seen Accuracy	99.3% $\pm$ 0.2%	99.5% $\pm$ 0.3%	98.9% $\pm$ 0.7%	99.5% $\pm$ 0.3%
Mean Client Unseen Accuracy	52.6% $\pm$ 7.1%	88.4% $\pm$ 3.5%	54.7% $\pm$ 5.1%	84.8% $\pm$ 2.6%
Training Time (s)	61.3 $\pm$ 8.5	86.4 $\pm$ 12.3	50.4 $\pm$ 7.9	69.7 $\pm$ 9.8
Communication Volume (MB)	365.2 $\pm$ 38.4	681.3 $\pm$ 68.7	285.3 $\pm$ 31.2	412.6 $\pm$ 42.3

As shown in Table 4, the full AFDL framework achieves the best overall balance between diagnostic performance and communication efficiency. Compared with M1, which removes the feature alignment mechanism, AFDL improves the server accuracy from 67.8% to 92.3% and the mean client unseen accuracy from 52.6% to 84.8%. This indicates that feature alignment plays a critical role in bridging the representation gap among heterogeneous client and server models. Specifically, the feature alignment loss constrains heterogeneous client and server representations into a more consistent latent space, thereby reducing the mismatch caused by different model architectures and data distributions. Without this constraint, the knowledge transferred from different clients becomes less consistent and more difficult to integrate effectively, leading to significant degradation in both server-side aggregation and client-side generalization to unseen fault categories.

Compared with M2, which replaces lightweight bidirectional knowledge distillation with complete parameter transmission, AFDL achieves comparable diagnostic performance while substantially reducing communication overhead. Although M2 obtains a slightly higher unseen accuracy, its communication volume reaches 681.3 MB, which is 65.1% higher than that of AFDL. Its training time also increases from 69.7 s to 86.4 s. This result demonstrates that complete parameter transmission can preserve more model information and may slightly improve unseen fault recognition, but it significantly increases the communication volume and training time. In contrast, AFDL transmits compact feature representations and soft probability distributions, reducing communication overhead while maintaining competitive diagnostic accuracy. Therefore, AFDL is more suitable for bandwidth-constrained satellite constellation scenarios.

Compared with M3, which removes the task-specific distillation phase, AFDL shows a significant improvement in client unseen fault diagnosis. The mean client unseen accuracy increases from 54.7% to 84.8%, while the server accuracy remains at a similar level. This confirms that the second-stage task-specific distillation mainly contributes to improving the generalization ability of local clients for unseen fault categories. Although this phase increases the training time and communication volume compared with M3, the substantial gain in unseen fault recognition demonstrates its necessity for practical constellation fault diagnosis.

Overall, the ablation results verify the effectiveness of the three core components of AFDL. The feature alignment mechanism improves heterogeneous knowledge fusion, the lightweight bidirectional distillation strategy reduces communication overhead compared with full-parameter transmission, and the task-specific distillation phase significantly enhances client-side generalization to unseen faults.

4.1.5. Comparison of Baseline Methods

We first compare AFDL with centralized learning in terms of training communication volume. In the centralized learning setting, clients are required to upload their local data

to the server for centralized model training, resulting in a total communication volume of 962.31 MB. In contrast, AFDL only exchanges model-related knowledge, including model updates, feature representations, and soft-label information, without transmitting raw telemetry data. Across 10 independent runs, the average communication volume of AFDL is 412.64 MB, corresponding to a 57.1% reduction compared with centralized learning. This result demonstrates that the proposed knowledge distillation strategy effectively reduces data transmission while preserving collaborative diagnostic learning across heterogeneous clients.

To further demonstrate the effectiveness of the proposed method, we compare it with several state-of-the-art federated learning approaches specifically designed for handling model and data heterogeneity. The selection of comparative methods was determined by the heterogeneous setting considered in this study. Unlike conventional federated learning scenarios that assume identical model architectures or homogeneous data distributions across clients, the proposed fault diagnosis task involves both model heterogeneity and data heterogeneity among different satellite platforms. Therefore, many classical federated learning methods, such as FedAvg and its direct variants, cannot be directly applied without imposing additional model unification or data homogenization assumptions, which would change the problem setting of this paper. For this reason, the comparative methods were selected from representative recent federated learning approaches that can handle heterogeneous models or heterogeneous knowledge transfer to some extent, including FedDF [27], FedKD [28], and FedGPD [29]. These methods cover the main feasible comparison categories for the proposed setting, including ensemble distillation, communication-efficient knowledge distillation, and prototype-based heterogeneous federated learning, thereby providing a relevant and fair comparison under the heterogeneous constellation fault diagnosis scenario considered in this work. To ensure a fair comparison, all baseline methods were implemented under the same experimental protocol as AFDL. They used the same client partition, client sampling strategy, local epochs, batch size, optimizer settings, training/test split, and seen/unseen fault division. For model capacity, each method used the same small-client, large-client, and server backbone architectures as those used in AFDL whenever applicable. For methods that do not explicitly distinguish seen and unseen fault categories, the same seen/unseen split was still retained for evaluation so that client-side generalization could be measured consistently. In addition, all methods were trained under the same maximum communication-round budget. Although AFDL adopts a two-stage training procedure, its total training budget was kept comparable to that of the baselines. The comparative results are summarized in Table 5.

Table 5. Comparison of baseline methods.

Indicator	FedDF	FedKD	FedGPD	AFDL
Server Accuracy Rate	81.1% $\pm$ 3.2%	69.2% $\pm$ 4.6%	78.4% $\pm$ 3.1%	92.3% $\pm$ 1.4%
Mean Client Seen Accuracy	99.3% $\pm$ 0.4%	99.1% $\pm$ 0.5%	98.7% $\pm$ 0.9%	99.5% $\pm$ 0.3%
Mean Client Unseen Accuracy	53.2% $\pm$ 5.7%	55.5% $\pm$ 6.9%	63.6% $\pm$ 4.3%	84.8% $\pm$ 2.6%
Training Time (s)	57.9 $\pm$ 2.8	49.3 $\pm$ 7.4	52.4 $\pm$ 7.6	69.7 $\pm$ 9.8
Communication Volume (MB)	286.5 $\pm$ 30.3	263.4 $\pm$ 28.7	315.2 $\pm$ 33.9	412.6 $\pm$ 42.3

As shown in Table 5, AFDL achieves the best overall diagnostic performance among all compared methods. Its server-side accuracy reaches 92.3% $\pm$ 1.4%, which is 11.2 percentage points higher than FedDF, the second-best method with 81.1% $\pm$ 3.2%. This result indicates that the proposed bidirectional distillation and feature-level alignment mechanisms enable the server to integrate heterogeneous client knowledge more effectively and learn more globally representative diagnostic features.

For client-side diagnosis, all methods maintain high seen-fault accuracy, but clear differences appear in unseen-fault recognition. AFDL achieves the highest mean client unseen accuracy of $84.8\% \pm 2.6\%$, outperforming FedGPD, FedKD, and FedDF by 21.2, 29.3, and 31.6 percentage points, respectively. This demonstrates that the task-specific distillation phase substantially improves client-side generalization to unseen fault categories. In contrast, FedDF mainly focuses on ensemble-based server fusion, FedKD relies on the quality of distilled knowledge transfer, and FedGPD alleviates non-IID feature shifts through global prototypes; however, these mechanisms do not explicitly optimize local clients for unseen fault recognition under heterogeneous satellite platforms.

In terms of efficiency, AFDL requires more training time and communication volume than the three baselines, with a training time of 69.7 ± 9.8 s and a communication volume of 412.6 ± 42.3 MB. Compared with FedGPD, AFDL increases training time by approximately 33.0% and communication volume by approximately 30.9%. This additional overhead is mainly introduced by the second-stage task-specific distillation and bidirectional knowledge transfer. Nevertheless, the overhead is accompanied by substantial improvements in server accuracy and unseen-fault recognition, which is acceptable for satellite fault diagnosis scenarios where diagnostic reliability is more critical than minimizing communication cost alone.

4.2. Public ESA Anomaly Dataset Experiment

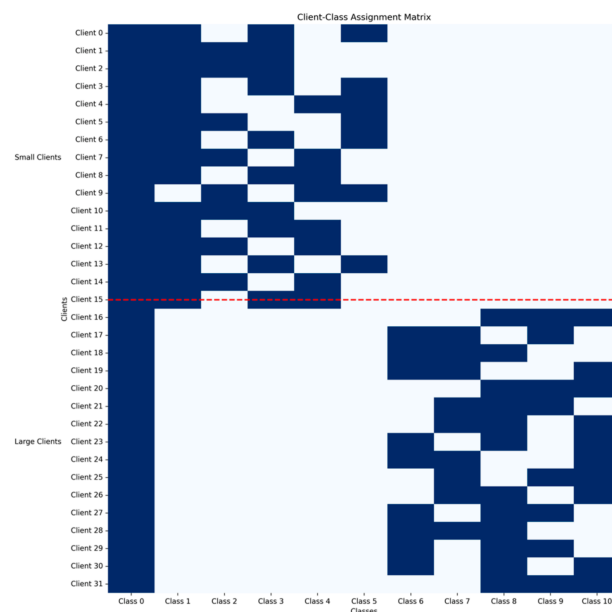
4.2.1. Experimental Data

The ESA Anomaly Dataset [30] represents the first large-scale annotated anomaly dataset derived from real satellite telemetry data, with annotations originating from three European Space Agency (ESA) space missions. The ESA dataset experiment follows the same evaluation protocol as the proprietary dataset experiment in Section 4.1, including the same random-seed strategy, metric reporting format, fairness protocol for baseline comparison, and communication-overhead measurement. Therefore, this section focuses on the dataset-specific configuration and the key comparative findings on the ESA dataset. This study utilizes ten months of data from Mission 1, covering the period from 1 January 2000 to 1 November 2000, at a sampling interval of 30 s, resulting in a total of 878,401 samples. Each sample comprises 76 dimensions of sensor channel measurements. Both the dataset and results are anonymized to prevent disclosure of mission-sensitive information such as channel names, timelines, or specific measurement types. The data is categorized into three major classes: 753,472 normal samples, 55,538 samples of Fault Type 1, and 69,391 samples of Fault Type 2. Each fault category corresponds to multiple combinations of channel anomalies: Fault Type 1 encompasses 86 distinct combinations involving channels 1 to 50, while Fault Type 2 includes 109 combinations spanning all channels. For the purpose of this study, we select the four most frequent combinations from each fault category and consolidate the remaining infrequent combinations into a fifth mode, resulting in a total of ten fault modes. This curated fault dataset is summarized in Table 6.

To simulate the structural differences and data isolation among various satellite models, this study configures small clients with sensor data from Channels 1 to 50, corresponding to Fault Category 1 as the fault data, while large clients are assigned sensor data from all channels, corresponding to Fault Category 2 as the fault data. Each client contains samples from the normal class and is assigned only 2 to 3 subcategories of fault data from their respective major fault category. The data distribution across clients is summarized in Figure 9.

Table 6. Summary of fault dataset categories.

	Fault Mode	Faulty Channel Combination	Number	Total
Normal			753,472	753,472
Fault Type 1	Fault Mode 1	Channel 22,30,31,38,39	20,560	69,391
	Fault Mode 2	Channel 15,22,23,31,39	18,561	
	Fault Mode 3	Channel 15,16,22,23,24,31,32,33,39,40	8139	
	Fault Mode 4	Channel 22,31,39	3909	
	Fault Mode 5	Others	18,222	
Fault Type 2	Fault Mode 6	Channel 16,24,32,33,40	21,595	55,538
	Fault Mode 7	Channel 16,24,32,33,40,47,48,49,50,51, 52,57,58,59,60,66	6953	
	Fault Mode 8	Channel 12,13,17,18,19,20,25,26,27,28,34,35, 36,37,41,42,43,44,45,46,47,48,49,50, 51,52	5638	
	Fault Mode 9	Channel 16,24,32,33,40,47,48,49,50,51,52,66	4287	
	Fault Mode 10	Others	22,703	

**Figure 9.** Client data distribution chart.

4.2.2. Parameter Settings

The experimental environment and methodology are identical to those as described in Section 4.1.2. Due to differences in the experimental dataset, specific parameters vary accordingly. The neural network architecture is shown in Table 7, and the training parameters are listed in Table 8. For the ESA dataset, the second training phase was terminated when the average local unseen-data accuracy of clients exceeded 75% while the local seen-data accuracy remained above 95%, or when the maximum number of training rounds was

reached. This threshold was lower than that used for the proprietary dataset because the ESA telemetry data contained more complex temporal patterns, higher-dimensional sensor channels, and more diverse anomaly combinations, making unseen fault recognition more difficult. Setting the threshold to 75% therefore provides a practical early-stopping criterion that balances unseen-fault generalization, training time, and communication overhead.

Table 7. Neural network architecture configuration.

	Small Client	Large Client	Server
Input Dimension	50	76	76
Feature Extractor	Linear (50, 256)	Linear (76, 512)	Linear (76, 512)
	Linear (256, 128)	Linear (512, 128)	Linear (512, 128)
Feature Adapter	Linear (128, 128)	Linear (128, 128)	Linear (128, 128)
Classifier	Linear (128, 11)	Linear (128, 11)	Linear (128, 11)
Dropout	0.3	0.3	0.3
Feature Dimension	128	128	128
Output Category	11	11	11
Total parameter count	107 K	183 K	183 K

Table 8. Training Parameter Settings.

Parameters	Phase 1	Phase 2	Instruction
Learning Round	50	50	Early Stopping with Conditions
Client selection per round	12/32	12/32	Random Uniform Sampling
Local training epoch	2	2	Client Local Update
Batch size	32	32	Data Loading Batch
Client learning rate	0.001	0.001	Adam Optimizer
Server learning rate	0.0005	0.0005	Adam Optimizer
Feature Alignment Weight	0.2	0.3	MSE Loss Coefficient
Distillation Loss Weight	0.6	Dynamic Adjustment	KL Divergence Loss Coefficient
Temperature Parameters τ	2.0	2.0	Knowledge Distillation Temperature
Weight Adjustment	Fixed	Dynamic Adjustment	Improvement on Unseen Classes

4.2.3. Experiment Result

The ESA dataset further verifies that AFDL maintains its advantage in unseen-fault generalization under a public satellite telemetry benchmark, although the more complex telemetry patterns lead to longer training time. Across the 10 independent runs, the proposed method completed the first training phase after 8 ± 3 communication rounds and completed the second training phase after 50 communication rounds. The final average seen-data accuracy on clients reached $99.39\% \pm 0.35\%$, while the unseen-data accuracy was $72.90\% \pm 3.28\%$. The server-side diagnostic accuracy reached $85.63\% \pm 1.92\%$, with an F1 score of $79.63\% \pm 2.47\%$ and a recall rate of $72.03\% \pm 3.76\%$. In terms of communication overhead, the total communication volume during training was 172.36 ± 22.42 MB, and the average experimental communication latency was 0.783 ± 0.135 s per round. The total training duration was 8967 ± 874 s, and the average client upload/download traffic was 2.81 ± 0.39 MB.

Among the 10 repeated experiments, the best-performing run was selected to visualize the actual diagnostic effect. In this run, the phase transition condition was met in round 6, after which the server ceased updating and acted as a fixed teacher to guide the clients through task-specific distillation on unseen fault categories. The final average seen accuracy on clients was 99.72%, with unseen accuracy at 76.17%. Server accuracy reached 87.14%, achieving an F1 score of 82.04% and a recall rate of 75.32%. The total communication volume during training was 161.87 MB, with an average experimental per-round training and communication time of 579 s. The total training duration was 8112 s, and the average client upload/download traffic was 2.43 MB. It should be noted that this study evaluates communication efficiency mainly from the perspective of transmitted data volume. Physical inter-satellite link effects, such as propagation delay, link acquisition delay, dynamic routing, and queueing delay, are not explicitly modeled and will be considered in future work. The training curves are depicted in Figure 10, and the server confusion matrix is shown in Figure 11.

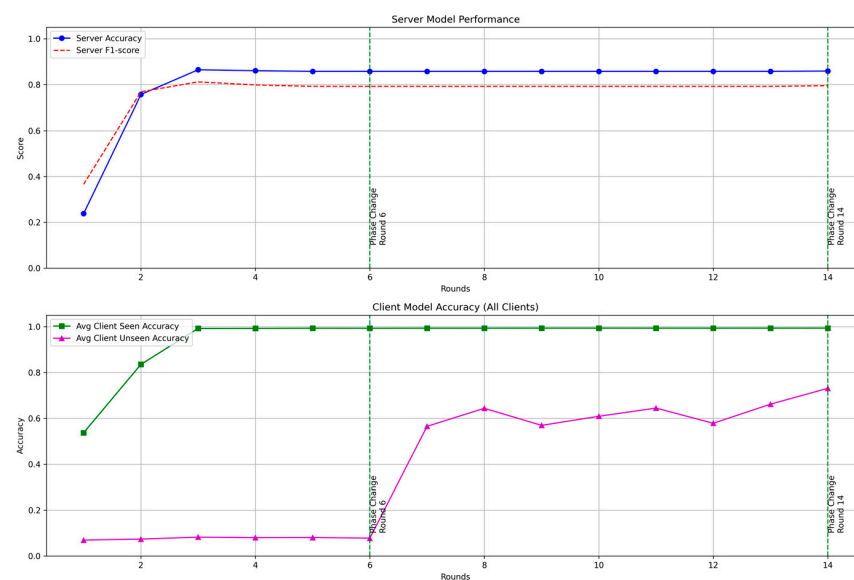


Figure 10. Training process curve.

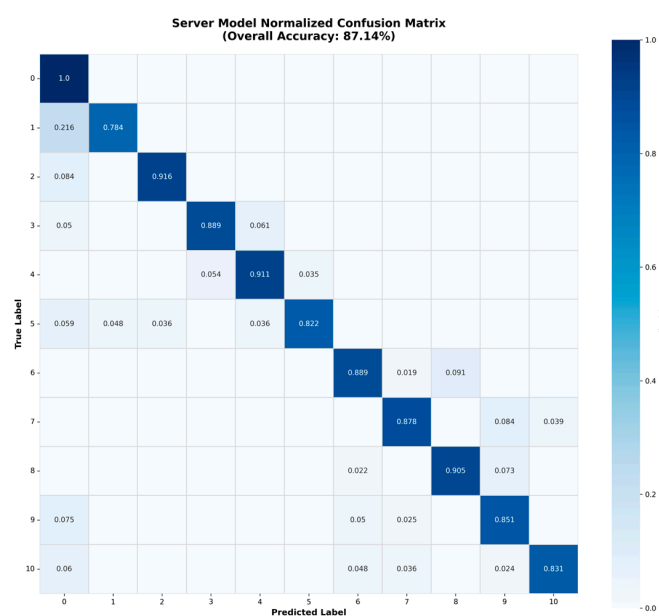


Figure 11. Server confusion matrix.

4.2.4. Ablation Experiments

To further verify the generalizability of the ablation conclusions, the same ablation variants were evaluated on the ESA dataset. The results show trends consistent with those on the self-collected dataset. The experimental results are shown in Table 9.

Table 9. Comparison of variant methods.

Indicator	M1	M2	M3	AFDL
Server Accuracy Rate	63.4% $\pm$ 3.8%	84.2% $\pm$ 2.6%	84.7% $\pm$ 2.3%	85.6% $\pm$ 1.9%
Mean Client Seen Accuracy	99.1% $\pm$ 0.5%	99.3% $\pm$ 0.4%	98.6% $\pm$ 0.8%	99.4% $\pm$ 0.3%
Mean Client Unseen Accuracy	46.8% $\pm$ 6.4%	74.1% $\pm$ 4.2%	51.6% $\pm$ 5.7%	72.9% $\pm$ 3.3%
Training Time (s)	7812 $\pm$ 806	11,264 $\pm$ 1187	6358 $\pm$ 742	8967 $\pm$ 874
Communication Volume (MB)	148.7 $\pm$ 19.3	286.41 $\pm$ 34.86	119.58 $\pm$ 16.24	172.36 $\pm$ 22.42

4.2.5. Comparison of Baseline Methods

Following the same comparison protocol as in Section 4.1.5, we further compare AFDL with centralized learning on the ESA dataset. The centralized learning method requires a total communication volume of 736.87 MB because local telemetry data need to be uploaded for server-side training. By contrast, AFDL only transmits compact training-related knowledge rather than raw data, resulting in an average communication volume of 172.36 MB over 10 independent runs. This corresponds to a 76.6% reduction in communication volume, further confirming the communication efficiency of AFDL on the public satellite telemetry benchmark. To further substantiate the generalization of AFDL, the proposed method is compared with FedDF, FedKD, and FedGPD on the ESA dataset, as summarized in Table 10.

Table 10. Comparison of baseline methods.

Indicator	FedDF	FedKD	FedGPD	AFDL
Server Accuracy Rate	75.7% $\pm$ 3.3%	64.8% $\pm$ 4.8%	71.6% $\pm$ 2.7%	85.6% $\pm$ 1.9%
Mean Client Seen Accuracy	99.2% $\pm$ 0.4%	99.1% $\pm$ 0.6%	98.7% $\pm$ 0.9%	99.4% $\pm$ 0.3%
Mean Client Unseen Accuracy	45.6% $\pm$ 6.1%	46.8% $\pm$ 6.4%	54.2% $\pm$ 5.3%	72.9% $\pm$ 3.3%
Training Time (s)	7423 $\pm$ 815	6810 $\pm$ 769	6972 $\pm$ 798	8967 $\pm$ 874
Communication Volume (MB)	126.84 $\pm$ 17.53	118.62 $\pm$ 15.94	132.27 $\pm$ 18.68	172.36 $\pm$ 22.42

The results show trends consistent with those observed on the proprietary dataset. AFDL achieves the highest server accuracy of 85.6% $\pm$ 1.9% and the highest mean client unseen accuracy of 72.9% $\pm$ 3.3%, outperforming the strongest baseline by 9.9 and 18.7 percentage points, respectively. Meanwhile, all methods maintain high seen-fault accuracy, indicating that the main advantage of AFDL lies in improving global server diagnosis and client-side unseen-fault generalization. Although AFDL requires higher training time and communication volume than the baseline methods, this additional overhead is acceptable considering its substantial improvement in unseen-fault recognition on the public ESA telemetry benchmark.

5. Conclusions

This paper addresses key challenges in constellation system fault diagnosis, including multimodal heterogeneity, communication constraints, and data scarcity, by proposing an Adaptive Federated Dual Knowledge Distillation (AFDL) framework. The approach establishes a multimodal heterogeneous federated architecture to effectively coordinate collaborative learning across satellite models of varying configurations. A lightweight bidirectional knowledge distillation strategy is introduced to significantly reduce commu-

nication overhead while maintaining diagnostic accuracy. Furthermore, a task-specific distillation phase is designed to integrate global guidance with client-specific optimization, substantially enhancing the generalization capability of local models. Experimental results demonstrate that AFDL outperforms existing federated learning methods in both global server-side performance and client-side accuracy in diagnosing unseen faults, while retaining high recognition rates for known fault types, indicating superior overall capability.

Despite these advancements, the current study still has several limitations. The experiments mainly evaluate communication efficiency from the perspective of transmitted data volume and experimental communication latency. Real inter-satellite communication environments are more complex, involving intermittent connectivity, time-varying topology, link acquisition delay, propagation delay, routing delay, queueing delay, and possible packet loss. These factors may affect the stability of federated aggregation and the timeliness of knowledge distillation in practical satellite constellations. Therefore, although the proposed method demonstrates promising communication efficiency under controlled experimental settings, its deployment in real satellite networks still requires further validation under dynamic topology and realistic link-delay conditions. Future work will focus on topology-aware asynchronous federation and delay-tolerant knowledge distillation mechanisms.

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

Funding: This research was funded by National Natural Science Foundation of China grant number 62022013. This research was funded by National Natural Science Foundation of China grant number U23B6002.

Data Availability Statement: The proprietary in-orbit satellite control system dataset is not publicly available due to institutional confidentiality restrictions. The ESA Anomaly Dataset used in this study is openly available at <https://github.com/kplabs-pl/ESA-ADB> (accessed on 23 April 2026) [30].

Acknowledgments: The authors would like to thank the Beijing Institute of Control Engineering for providing the in-orbit satellite telemetry data used in this study. During the preparation of this manuscript, the authors used ChatGPT 5 for language polishing and formatting. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

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

References

1. Kulu, E. Satellite constellations—2024 survey, trends and economic sustainability. In Proceedings of the International Astronautical Congress, IAC, Milan, Italy, 14–18 October 2024; pp. 14–18.
2. Hedayati, M.S.; Barzegar, A.; Rahimi, A. Fault diagnosis and prognosis of satellites and unmanned aerial vehicles: A review. *Appl. Sci.* **2024**, *14*, 9487. [\[CrossRef\]](#)
3. Teng, F.; Zhu, Y.; Zhang, E.; Hu, X.; Sun, Q.; Feng, L. A knowledge service framework for fault diagnosis of low-earth orbit satellite constellation. In *2023 IEEE International Conference on Web Services (ICWS)*; IEEE: New York, NY, USA, 2023; pp. 669–676.
4. Liu, C.; Chen, L.; Ding, J.; Shangguan, D. Modeling of satellite constellation in modelica and a PHM system framework driven by model data hybrid. *Electronics* **2022**, *11*, 2155. [\[CrossRef\]](#)
5. Han, Y.; Fan, W.; Han, F.; Cheng, H. Low-Orbit Giant Constellation Fault Propagation and Fault Source Detection Algorithm. In *2024 43rd Chinese Control Conference (CCC)*; IEEE: New York, NY, USA, 2024; pp. 5002–5007.
6. Zhu, Z.; Lei, Y.; Qi, G.; Chai, Y.; Mazur, N.; An, Y.; Huang, X. A review of the application of deep learning in intelligent fault diagnosis of rotating machinery. *Measurement* **2023**, *206*, 112346. [\[CrossRef\]](#)
7. Wang, C.; Sun, Y.; Wang, X. Image deep learning in fault diagnosis of mechanical equipment. *J. Intell. Manuf.* **2024**, *35*, 2475–2515.

8. Kumar, D.; Addula, S.R.; Lind, M.; Brown, S.; Odion, S. AI-Driven Hybrid Deep Learning and Swarm Intelligence for Predictive Maintenance of Smart Manufacturing Robots in Industry 4.0. *Electronics* **2026**, *15*, 715. [\[CrossRef\]](#)
9. Ding, Y.; Ma, L.; Ma, J.; Suo, M.; Tao, L.; Cheng, Y.; Lu, C. Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. *Adv. Eng. Inform.* **2019**, *42*, 100977. [\[CrossRef\]](#)
10. Li, Z.; Jiang, H.; Wang, X. A novel reinforcement learning agent for rotating machinery fault diagnosis with data augmentation. *Reliab. Eng. Syst. Saf.* **2025**, *253*, 110570. [\[CrossRef\]](#)
11. Zhang, C.; Wang, Y.; You, X. Fault diagnosis in rotating machinery with discretized signal representation leveraging large language models. *Appl. Soft Comput.* **2025**, *189*, 114487. [\[CrossRef\]](#)
12. Xu, C.; Wang, Z.; Jin, Y.; Nong, W. An adaptive industrial large language model for mechanical fault diagnosis under variable operating conditions. *Adv. Eng. Inform.* **2026**, *74*, 104821. [\[CrossRef\]](#)
13. Wen, J.; Zhang, Z.; Lan, Y.; Cui, Z.; Cai, J.; Zhang, W. A survey on federated learning: Challenges and applications. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 513–535. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Boobalan, P.; Ramu, S.P.; Pham, Q.V.; Dev, K.; Pandya, S.; Maddikunta, P.K.R.; Gadekallu, T.R.; Huynh-The, T. Fusion of federated learning and industrial Internet of Things: A survey. *Comput. Netw.* **2022**, *212*, 109048. [\[CrossRef\]](#)
15. Younis, R.; Fisichella, M. FLY-SMOTE: Re-balancing the non-IID IoT edge devices data in federated learning system. *IEEE Access* **2022**, *10*, 65092–65102. [\[CrossRef\]](#)
16. Zhang, W.; Li, X.; Ma, H.; Luo, Z.; Li, X. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. *Knowl.-Based Syst.* **2021**, *213*, 106679. [\[CrossRef\]](#)
17. Kang, S.; Sun, Y.; Li, X.; Wang, Y.; Wang, Q.; Liang, X. Unsupervised fault diagnosis method for rolling bearings based on federated universal domain adaptation. *Eng. Appl. Artif. Intell.* **2025**, *162*, 112518. [\[CrossRef\]](#)
18. Yan, P.; Hu, Y.; Wen, W.; Hsu, L.-T. Multiple Faults Isolation for Multi-Constellation GNSS Positioning through Incremental Expansion of Consistent Measurements. *IEEE Sens. J.* **2025**, *25*, 6967–6981. [\[CrossRef\]](#)
19. Meng, Q.; Liu, J.; Zeng, Q.; Feng, S.; Xu, R. Impact of one satellite outage on ARAIM depleted constellation configurations. *Chin. J. Aeronaut.* **2019**, *32*, 967–977. [\[CrossRef\]](#)
20. Huang, G.; Xu, C.; Zhao, J.; Song, D. Bayesian fault-tolerant protection level for multi-constellation navigation from integrity perspective. *Aerosp. Sci. Technol.* **2022**, *130*, 107954. [\[CrossRef\]](#)
21. Zhang, K.; Song, X.; Zhang, C.; Yu, S. Challenges and future directions of secure federated learning: A survey. *Front. Comput. Sci.* **2022**, *16*, 165817.
22. Dai, M.; Xu, A.; Huang, Q.; Zhang, Z.; Lin, X. Vertical federated DNN training. *Phys. Commun.* **2021**, *49*, 101465. [\[CrossRef\]](#)
23. Guan, J.; Cai, J.; Bai, H.; You, I. Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems. *Int. J. Mach. Learn. Cybern.* **2021**, *12*, 3351–3365. [\[CrossRef\]](#)
24. Jiang, S.; Wang, B.; Zhang, X.; Jiang, Y.; Liu, S.; Zhao, Z.; Li, R.; Chen, X. ObsBattery: Position-Aware Federated Learning with Dueling DQN Clustering and Training Adaptation for Satellite Battery Prediction. *Electronics* **2025**, *14*, 4697. [\[CrossRef\]](#)
25. Zhang, P.; Wang, C.; Jiang, C.; Han, Z. Deep reinforcement learning assisted federated learning algorithm for data management of IIoT. *IEEE Trans. Ind. Inform.* **2021**, *17*, 8475–8484. [\[CrossRef\]](#)
26. Lu, S.; Gao, Z.; Xu, Q.; Jiang, C.; Zhang, A.; Wang, X. Class-imbalance privacy-preserving federated learning for decentralized fault diagnosis with biometric authentication. *IEEE Trans. Ind. Inform.* **2022**, *18*, 9101–9111. [\[CrossRef\]](#)
27. Lin, T.; Kong, L.; Stich, S.U.; Jaggi, M. Ensemble distillation for robust model fusion in federated learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 2351–2363.
28. Wu, C.; Wu, F.; Lyu, L.; Huang, Y.; Xie, X. Communication-efficient federated learning via knowledge distillation. *Nat. Commun.* **2022**, *13*, 2032. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Wu, S.; Chen, J.; Nie, X.; Wang, Y.; Zhou, X.; Lu, L.; Peng, W.; Nie, Y.; Menhaj, W. Global prototype distillation for heterogeneous federated learning. *Sci. Rep.* **2024**, *14*, 12057. [\[CrossRef\]](#) [\[PubMed\]](#)
30. Kotowski, K.; Haskamp, C.; Andrzejewski, J.; Ruszczak, B.; Nalepa, J.; Lakey, D.; Collins, P.; Kolmas, A.; Bartesaghi, M.; Martinez-Heras, J.; et al. European space agency benchmark for anomaly detection in satellite telemetry. *arXiv* **2024**, arXiv:2406.17826.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.