mathematics

MDPI

Article

Privacy-Preserving U-Shaped Split Federated Learning for Space–Air–Ground–Sea Integrated Networks

Xin Sun, Tingting Yang * and Xiufeng Zhang

The Navigation College, Dalian Maritime University, Dalian 116026, China; sunxin1996@dlmu.edu.cn (X.S.); zxfdmu@dlmu.edu.cn (X.Z.)

* Correspondence: yangtingting@dlmu.edu.cn; Tel.: +86-188-4268-3387

Abstract

Federated learning enables privacy-preserving distributed intelligence but faces challenges in balancing computation, communication, and privacy in heterogeneous networks. To address these issues, this paper proposes a privacy-preserving U-shaped split federated learning (USFL) framework for space–air–ground–sea integrated networks. The proposed architecture combines split learning and federated learning in a U-shaped structure, ensuring that both raw data and labels remain localized at client devices. In addition, a differential privacy mechanism is introduced to perturb intermediate features during transmission, enhancing resistance to inference attacks. A mathematical framework is established to model the learning process under resource constraints, and the convergence behavior and privacy loss are theoretically analyzed. Experimental results on the SeaShips dataset demonstrate that the proposed method achieves competitive accuracy compared with centralized and existing distributed approaches, while reducing communication overhead and improving privacy protection. These results validate the effectiveness of the proposed framework for secure and efficient distributed learning in complex network environments.

Keywords: federated learning; split learning; privacy-preserving learning; differential privacy; edge intelligence; resource-constrained networks; heterogeneous networks

MSC: 68T07

1. Introduction

With the rapid development of artificial intelligence and communication technologies, distributed collaborative learning has become a fundamental paradigm for enabling intelligent services in large-scale networked systems [1]. Federated learning (FL), as a representative distributed learning framework, allows multiple clients to collaboratively train machine learning models without sharing raw data, thereby providing an effective solution for privacy-sensitive applications [2,3]. Owing to its data-localization property, FL has been widely adopted in edge intelligence, Internet of Things (IoT), and intelligent transportation systems [4–6]. However, in practical deployments, especially in heterogeneous and resource-constrained environments, conventional FL still faces significant challenges.

In complex multi-tier network architectures such as space–air–ground–sea integrated networks (SAGSINs), the system typically consists of satellites, aerial platforms, terrestrial base stations, and maritime devices. These networks are characterized by wide coverage, heterogeneous node types, and dynamic communication conditions [7]. Meanwhile, different nodes exhibit substantial disparities in computational capability, storage capacity,

Check for updates

Academic Editor: Haozhao Wang

Received: 25 March 2026
Revised: 15 April 2026
Accepted: 16 April 2026
Published: 18 April 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) license.

and energy consumption. Under such conditions, achieving efficient, stable, and privacy-preserving distributed learning remains a critical challenge.

Traditional FL methods generally assume that each client is capable of training a full model locally, which is impractical for resource-constrained devices such as unmanned surface vehicles (USVs) [8,9] and unmanned aerial vehicles (UAVs) [10,11]. On the one hand, deep neural networks impose high computational and energy demands on edge devices [12,13]. On the other hand, frequent transmission of model parameters leads to considerable communication overhead, further burdening the network. To address these issues, split learning (SL) has been proposed, which partitions a model into client-side and server-side segments, allowing computationally intensive tasks to be offloaded to edge servers or the cloud [14,15]. However, SL requires the transmission of intermediate features between clients and servers. These intermediate representations often contain rich semantic information, making them vulnerable to feature inversion and inference attacks, thus introducing new privacy risks.

Recent studies have attempted to combine FL [16] and SL into hybrid paradigms, such as split federated learning [17,18], to simultaneously exploit distributed optimization and model partitioning. Although these approaches alleviate some of the limitations of resource-constrained learning, they still suffer from several fundamental issues. First, most existing methods are designed in an empirical manner, lacking a systematic mathematical modeling framework under heterogeneous resource constraints, which limits the theoretical understanding of the interplay among computation, communication, and privacy [19–21]. Second, existing privacy-preserving mechanisms mainly focus on gradients or model parameters, while the privacy risks associated with intermediate feature transmission are not sufficiently addressed. Third, there is a lack of rigorous analysis on the trade-off between privacy protection and learning performance, particularly in terms of convergence behavior and privacy loss [22,23].

To tackle the above challenges, this paper proposes a privacy-preserving U-shaped split federated learning framework for heterogeneous and resource-constrained environments. The proposed approach integrates the advantages of FL and SL through a U-shaped closed-loop architecture, where both forward and backward propagation originate and terminate at the client side. This design ensures that raw data and label information always remain local, thereby effectively reducing the risk of information leakage. In addition, to enhance the security of intermediate feature transmission, we introduce a differential privacy-based noise perturbation mechanism, which randomizes intermediate representations and mitigates inference attacks.

From a mathematical perspective, we further establish a formal modeling framework for the proposed learning system under heterogeneous and resource-constrained conditions. Specifically, we formulate the learning process by incorporating computational cost, communication overhead, and privacy constraints into a unified model. We analyze the convergence behavior of the proposed method under stochastic noise injection and derive bounds on the privacy loss based on differential privacy theory. Furthermore, we characterize the impact of noise perturbation on model accuracy and training stability, revealing the intrinsic trade-off between privacy protection and learning utility.

The main contributions of this paper are summarized as follows:

- A novel U-shaped split federated learning framework is proposed, which integrates split learning and federated learning into a unified architecture, ensuring that both raw data and label information remain localized at client devices.
- A differential privacy-based mechanism is introduced at the intermediate feature level, which enhances privacy protection against inference attacks during feature transmission.

- A unified mathematical modeling framework is developed to jointly characterize computation, communication, and privacy, along with theoretical analysis on convergence behavior and privacy–utility trade-offs.
- Extensive experiments demonstrate that the proposed method achieves improved performance in terms of accuracy, communication efficiency, and privacy preservation compared with existing approaches.

The remainder of this paper is organized as follows. Section 2 introduces the mathematical system model. Section 3 presents the proposed method and privacy mechanisms. Section 4 provides theoretical analysis. Section 5 reports experimental results, and Section 6 concludes the paper.

2. System Model and Problem Formulation

2.1. Space–Air–Ground–Sea Integrated Network Architecture

The collaborative learning framework considered in this paper is deployed over a space–air–ground–sea integrated network (SAGSIN), which is a multi-layer heterogeneous architecture that integrates communication and computation resources across different domains [24]. As illustrated in Figure 1, the SAGSIN consists of four hierarchical layers, namely the space layer, air layer, ground layer, and sea layer, which are interconnected via multi-hop communication links to form a unified cross-domain network for distributed intelligence.

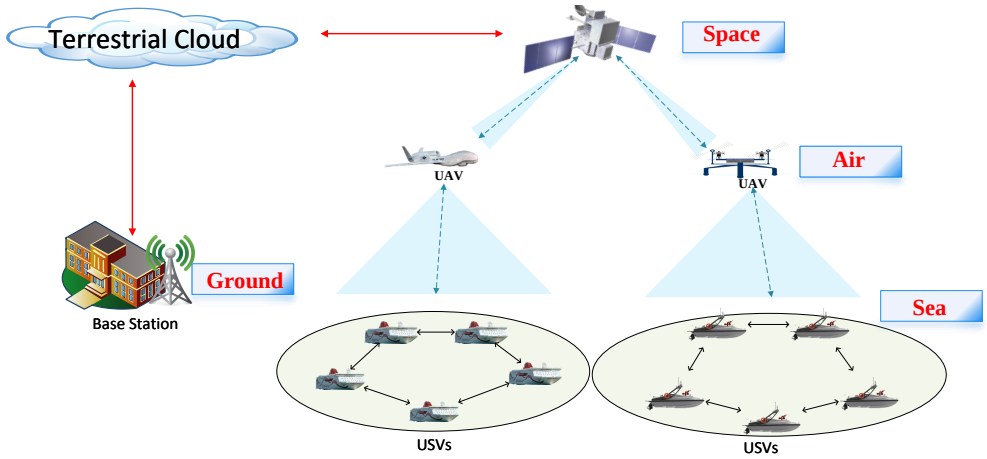


Figure 1. Architecture of the space–air–ground–sea integrated network.

In this architecture, different layers undertake distinct functional roles. The space layer, composed of satellites, provides wide-area coverage and long-distance communication support. The air layer, consisting of UAVs, serves as flexible relay nodes and edge computing units, enabling data aggregation and task offloading. The ground layer, including base stations and terrestrial cloud servers, is responsible for global coordination, model aggregation, and resource scheduling. The sea layer comprises USVs and other terminal devices, which perform data sensing, local inference, and preliminary computation tasks. Due to the inherent disparities in computational capability, communication bandwidth, and energy constraints across different layers, the network exhibits a highly heterogeneous resource structure.

For mathematical modeling, let $\mathcal{K} = 1, 2, \ldots, K$ denote the set of client nodes (e.g., USVs), $\mathcal{A} = 1, 2, \ldots, A$ denote the set of aerial nodes (e.g., UAVs), and $\mathcal{G} = 1, 2, \ldots, G$ denote the set of ground servers. Each client $k \in \mathcal{K}$ possesses a local dataset $\mathcal{D}_k$, where the data distributions across clients are generally non-independent and identically distributed (non-IID). During the collaborative learning process, intermediate outputs generated by

clients are transmitted to aerial nodes and subsequently forwarded to ground servers for further computation or global aggregation.

In the communication model, wireless links are established between clients and aerial nodes with transmission rate denoted by $R_{k,a}$, while the link between aerial nodes and ground servers is characterized by rate $R_{a,g}$. The multi-hop transmission process introduces communication latency, which depends on bandwidth constraints and channel conditions. In the computation model, the processing capabilities of clients, aerial nodes, and ground servers are denoted by $f_k^{(c)}$, $f_a^{(a)}$, and $f_g^{(g)}$, respectively, capturing the heterogeneity of computational resources across different layers. Since client devices are typically resource-constrained, they cannot independently execute full model training and therefore rely on computation offloading to more capable nodes.

Based on the above system architecture, the distributed learning process involves three fundamental factors: computational cost, communication overhead, and privacy risk [25]. Computational cost arises from forward and backward propagation across distributed nodes; communication overhead is incurred by transmitting intermediate representations between layers; and privacy risk stems from the potential leakage of sensitive information embedded in intermediate features during transmission. Therefore, designing an efficient collaborative learning framework in SAGSIN requires achieving a careful balance among computation efficiency, communication cost, and privacy preservation.

Compared with traditional centralized maritime systems [26], SAGSIN enables computation and decision-making to be shifted closer to data sources, thereby reducing communication latency and system burden. However, multi-node collaboration also introduces new challenges. On the one hand, intermediate data transmitted over wireless links may be intercepted or exploited by adversaries, leading to privacy leakage. On the other hand, the absence of effective knowledge-sharing mechanisms among distributed nodes may result in redundant computations and inefficient resource utilization. These challenges motivate the development of a distributed learning framework that simultaneously ensures efficiency, scalability, and privacy preservation in heterogeneous network environments.

Building upon the above system model, the next subsection introduces the proposed U-shaped split federated learning architecture, which is specifically designed for collaborative learning under heterogeneous and resource-constrained conditions. A formal mathematical formulation will then be developed based on this architecture.

2.2. The U-Shaped Split Federated Learning Architecture

To address the challenges of heterogeneous resource constraints and privacy preservation in space–air–ground–sea integrated networks, a privacy-preserving U-shaped split federated learning (USFL) architecture is developed based on the aforementioned system model. This architecture integrates the advantages of federated learning and split learning, enabling collaborative model training while maintaining data locality and supporting computation offloading across distributed nodes. As illustrated in Figure 2, the proposed framework consists of clients, edge servers, and a parameter server, where the training process is realized through the transmission of three types of information flows, namely intermediate features, gradients, and model parameters, corresponding to forward propagation, backward propagation, and global aggregation, respectively.

Consider a deep neural network model $F(\cdot)$ with L layers. To enable distributed execution, the model is decomposed into three sequential sub-models such that

$$F(x) = F^{(3)}(F^{(2)}(F^{(1)}(x))). \tag{1}$$

Here, $F^{(1)}(\cdot)$ is deployed on the client side to perform shallow feature extraction, $F^{(2)}(\cdot)$ is executed on the edge server to handle computation-intensive intermediate layers,

and $F^{(3)}(\cdot)$ is returned to the client to complete final prediction and loss computation. This decomposition forms a U-shaped execution path, i.e., client–server–client, ensuring that the training process is locally closed at the client side and preventing the exposure of raw data and labels [27].

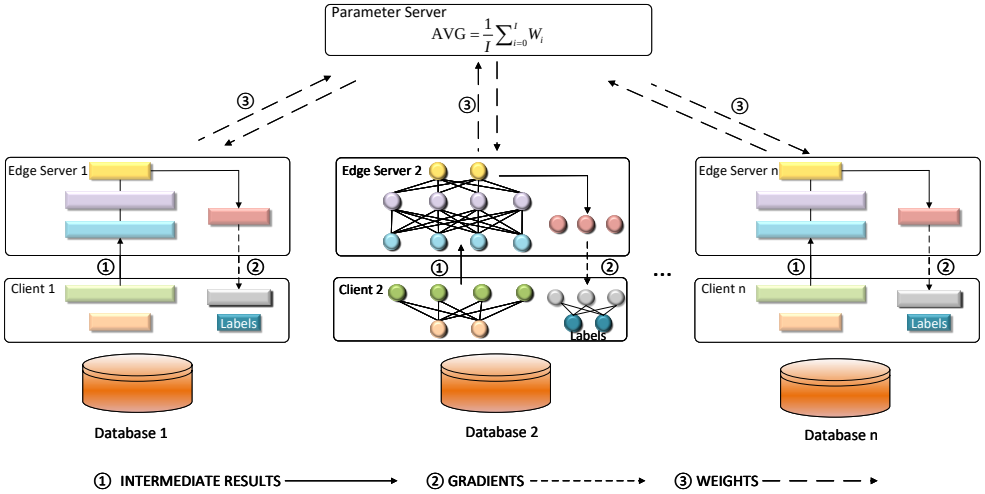


Figure 2. The U-shaped split federated learning framework.

During forward propagation, for each client $k \in \mathcal{K}$ with input data x_k, the client first computes intermediate representations as $h_k = F^{(1)}(x_k)$. To enhance privacy during transmission, a perturbation mechanism is applied to the intermediate features, yielding $\tilde{h}_k = h_k + \mathcal{N}(0, \sigma^2)$, which are then transmitted to the edge server. The server performs intermediate computation as $z_k = F^{(2)}(\tilde{h}_k)$ and sends the results back to the client. The client subsequently computes the final prediction $\hat{y}_k = F^{(3)}(z_k)$. Throughout this process, raw data remain strictly local, while only perturbed intermediate representations are communicated across nodes.

In the backward propagation phase, the client computes the loss function $\mathcal{L}_k = \ell(\hat{y}_k, y_k)$ based on the ground-truth label y_k and initiates gradient backpropagation. The gradient with respect to the intermediate representation is computed as $\nabla z_k = \partial \mathcal{L}_k / \partial z_k$ and transmitted to the server, which updates the intermediate model parameters $\theta^{(2)}$. The gradients are then propagated back to the client to update the front-end model parameters $\theta^{(1)}$. Since labels are never exposed to the server, this mechanism effectively mitigates label leakage and inference attacks.

At the end of each training round, local model parameters from all participating clients are uploaded to the parameter server for global aggregation. Let $\theta_k^{(t)}$ denote the model parameters of client k at round t. The global model is updated as

$$\theta^{(t+1)} = \sum_{k=1}^{K} \frac{|\mathcal{D}_k|}{\sum_j |\mathcal{D}_j|} \theta_k^{(t)}. \tag{2}$$

The aggregated model is then redistributed to all clients, enabling knowledge sharing and synchronization across distributed nodes.

Overall, the proposed U-shaped split federated learning architecture establishes a unified collaborative learning framework in heterogeneous network environments. By combining model partitioning for computation offloading, federated aggregation for global optimization, and feature perturbation for privacy protection, the framework achieves an effective balance among computational efficiency, communication overhead, and privacy preservation. This architecture provides a solid foundation for the subsequent mathematical modeling and optimization formulation.

2.3. Problem Formulation and Mathematical Modeling

Based on the aforementioned system model and the U-shaped split federated learning architecture, we further establish a formal mathematical framework to characterize the computation and communication processes in distributed learning over space–air–ground–sea integrated networks.

Let $\mathcal{K} = 1, 2, \ldots, K$ denote the set of participating clients, where each client k possesses a local dataset $\mathcal{D} \times k$ with size $|\mathcal{D} \times k|$. Under the federated learning paradigm, the global training objective can be formulated as a weighted empirical risk minimization problem:

$$\min_{\theta} F(\theta) = \sum_{k=1}^{K} \frac{|D_k|}{\sum_{j=1}^{K} |D_j|} \mathbb{E}_{(x_k, y_k) \sim D_k} [\ell(F(x_k; \theta), y_k)] \tag{3}$$

where θ denotes the global model parameters and $\ell(\cdot)$ represents the loss function.

Under the U-shaped split federated learning architecture, the training process is jointly executed by clients and edge servers. The system overhead primarily consists of computation latency and communication latency. From the perspective of computation, clients execute both the front-end and back-end sub-models, while the intermediate layers are processed by edge servers. Accordingly, the computation latency for client k can be expressed as

$$T_k^{\text{comp}} = \frac{C_k^{(1)} + C_k^{(3)}}{f_k^{(c)}} + \frac{C_k^{(2)}}{f_a^{(a)}}, \tag{4}$$

where $C_k^{(1)}$, $C_k^{(2)}$, and $C_k^{(3)}$ denote the computational complexity of the front-end, intermediate, and back-end sub-models, respectively, while $f_k^{(c)}$ and $f_a^{(a)}$ represent the computational capabilities of the client and the edge server.

From the communication perspective, the U-shaped architecture requires bidirectional data exchange between clients and servers, including intermediate features, inference outputs, and gradients. The corresponding communication latency is given by

$$T_k^{\text{comm}} = \frac{S_k^{(h)}}{R_{k,a}} + \frac{S_k^{(z)}}{R_{a,k}} + \frac{S_k^{(g)}}{R_{k,a}}, \tag{5}$$

where $S_k^{(h)}$, $S_k^{(z)}$, and $S_k^{(g)}$ denote the sizes of transmitted intermediate features, returned outputs, and gradients, respectively, and $R_{k,a}$ represents the communication rate between the client and the edge node.

In addition, during the federated aggregation stage, local model parameters are uploaded from clients to the parameter server for global model updating. The corresponding communication cost is expressed as

$$T^{\text{agg}} = \sum_{k=1}^{K} \frac{S_k^{(\theta)}}{R_{k,g}}, \tag{6}$$

where $S_k^{(\theta)}$ denotes the size of model parameters and $R_{k,g}$ represents the communication rate between the client and the ground server.

By combining the above components, the total system latency can be formulated as

$$T^{\text{total}} = \sum_{k=1}^{K} \left(T_k^{\text{comp}} + T_k^{\text{comm}} \right) + T^{\text{agg}}. \tag{7}$$

Accordingly, the collaborative learning process can be formulated as the following optimization problem $\min T^{\text{total}}$.

This optimization problem is subject to system resource constraints. Specifically, the computational capabilities of nodes are bounded by hardware limitations, i.e., $f_k^{(c)} \leq f_k^{\max}$ and $f_a^{(a)} \leq f_a^{\max}$. The communication rates are constrained by channel conditions, i.e., $R_{k,a} \leq R_{k,a}^{\max}$ and $R_{k,g} \leq R_{k,g}^{\max}$. Moreover, the data transmission sizes are non-negative and determined by the model structure, i.e., $S_k^{(h)}, S_k^{(g)}, S_k^{(\theta)} \geq 0$.

Compared with conventional split learning and federated learning frameworks, the proposed U-shaped architecture introduces a client–server–client execution path, which fundamentally changes the information flow in distributed training.

In traditional split learning, label information is typically exposed to the server during the final prediction stage, which may lead to potential privacy leakage. In contrast, the U-shaped structure ensures that both raw data and labels remain strictly local at the client side, thereby eliminating label leakage risks from a structural perspective.

Moreover, by returning the back-end model to the client, the proposed design forms a closed-loop training process, which enhances the security of intermediate representations and reduces the attack surface for inference attacks.

From a system perspective, the U-shaped architecture enables flexible computation offloading while maintaining local control of sensitive information, making it particularly suitable for heterogeneous and resource-constrained environments. Therefore, the proposed design achieves a more effective balance among computation, communication, and privacy compared with existing FL, SL, and SFL frameworks.

Through this formulation, the training process of U-shaped split federated learning in space–air–ground–sea integrated networks is unified as a latency-minimization optimization problem. This model provides a theoretical foundation for subsequent algorithm design and performance analysis.

3. U-Shaped Split Federated Learning Method

3.1. Model Splitting Strategy

In heterogeneous space–air–ground–sea integrated networks (SAGSINs), the model splitting strategy plays a critical role in determining computational load distribution, communication efficiency, and overall learning performance. Although the U-shaped learning framework has been introduced in the previous section, how to design an appropriate model partition under resource constraints remains a key challenge.

As illustrated in Figure 3, a deep neural network is decomposed into multiple consecutive layers and distributed across clients and edge servers following a U-shaped structure [28]. Specifically, the client executes the front-end and back-end submodels, while the edge server is responsible for processing the intermediate computationally intensive layers. This architecture forms a closed-loop execution path of "client–server–client", where both forward and backward propagations originate and terminate at the client side.

In distributed environments, different layers of a deep model exhibit distinct characteristics in terms of computational complexity and feature representation. The shallow layers are mainly responsible for low-level feature extraction, with relatively low computational cost and moderate output dimensionality, making them suitable for deployment on resource-constrained clients. In contrast, intermediate layers capture high-level semantic representations and typically involve higher computational complexity and stronger inter-layer dependencies, thus being more appropriate for execution at edge servers. The back-end layers are primarily used for classification and decision-making, and are usually lightweight, enabling their deployment at the client side for local inference and loss computation.

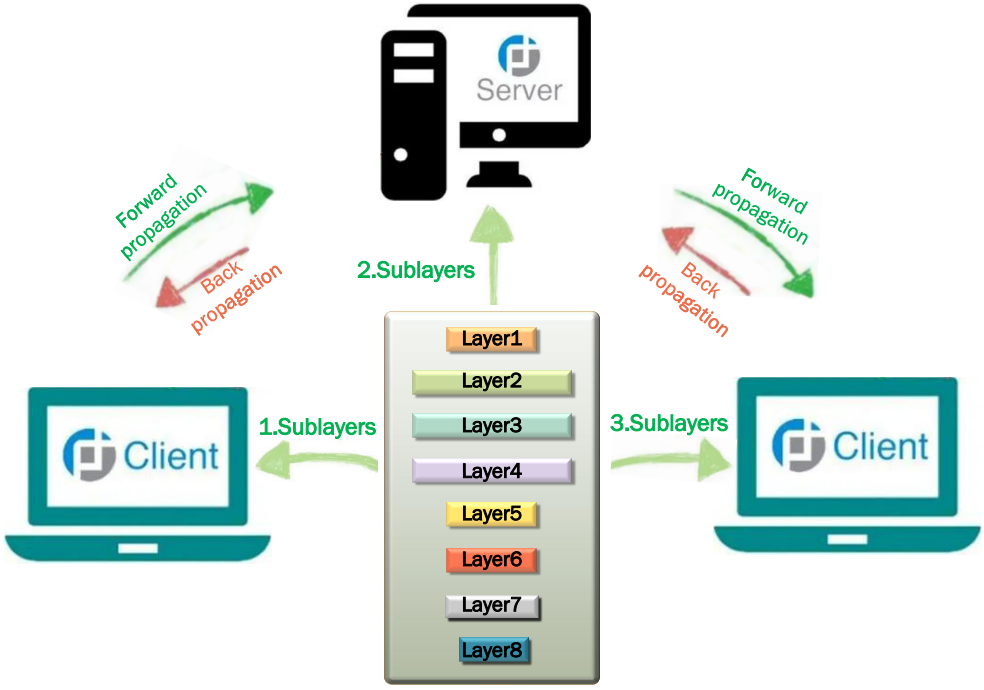


Figure 3. USFL Layer-wise Partition Strategy under ResNet18.

To provide a concrete instantiation of the proposed strategy, we consider a representative deep convolutional model, i.e., ResNet [29]. Due to its modular residual structure and balanced depth, ResNet is well suited for partitioned execution in distributed environments. Following the characteristics of layer-wise functionality, the model can be divided into three segments: (i) shallow convolutional layers for initial feature extraction, (ii) intermediate residual blocks for high-level representation learning, and (iii) final fully connected layers for classification. Accordingly, the shallow and final layers are deployed at the client side, while the residual blocks are offloaded to the edge server. This partitioning aligns with the general principle of assigning computation-intensive components to resource-rich nodes while preserving lightweight operations locally.

Under this splitting strategy, the intermediate feature serves as the key information carrier across distributed nodes, which can be expressed as

$$h_k = F^{(1)}(x_k), \tag{8}$$

where its dimensionality and semantic richness are jointly determined by the chosen split position. As shown in Figure 3, different splitting configurations correspond to different layer allocations, which directly affect the size of transmitted features and the associated communication cost. A shallow split increases the computational burden on the client while reducing transmission overhead, whereas a deeper split offloads more computation to the server at the expense of transmitting higher-dimensional features. Therefore, the model splitting strategy can be essentially viewed as a trade-off between computational complexity and communication overhead.

Furthermore, the U-shaped structure relocates the back-end model to the client side, forming a closed-loop training process locally. This design prevents label information from being exposed to the server and reduces the risk of inference attacks from a structural perspective. Considering that the back-end model is typically lightweight, the additional computational overhead introduced at the client remains acceptable, thereby achieving a balance between system efficiency and privacy preservation.

In summary, the proposed model splitting strategy jointly considers layer-wise model characteristics and system resource constraints, enabling an effective trade-off between computation offloading and communication efficiency. Meanwhile, the U-shaped closed-loop structure enhances privacy protection by design, and establishes a foundation for subsequent privacy-preserving mechanisms based on intermediate feature transmission.

3.2. Differential Privacy Mechanism over Intermediate Features

In the U-shaped split learning framework, intermediate features serve as the primary information carrier transmitted between clients and edge servers, and their representation directly affects the privacy security of the system. Although raw data remain strictly local, intermediate features may still encode sensitive semantic information of the input data, thereby exposing the system to potential feature inversion and inference attacks. Therefore, it is essential to introduce a rigorous privacy-preserving mechanism during feature transmission.

Let $F^{(1)}(\cdot)$ denote the feature mapping function at the client side. The generated intermediate feature can be expressed as h_k. To protect privacy, a stochastic perturbation mechanism is applied before transmission by injecting Gaussian noise into the feature representation, yielding

$$\tilde{h}_k = h_k + \mathcal{N}(0, \sigma^2 I). \tag{9}$$

within the differential privacy framework, the privacy guarantee depends on the sensitivity of the feature mapping. The ℓ_2-sensitivity of $F^{(1)}(\cdot)$ is defined as

$$\Delta = \max(x, x') |F^{(1)}(x) - F^{(1)}(x')|_2, \tag{10}$$

where x and x' are adjacent inputs. To relate input perturbations to feature variations, we assume that $F^{(1)}(\cdot)$ is Lipschitz continuous, i.e., there exists a constant L such that

$$|F^{(1)}(x) - F^{(1)}(x')|_2 \leq L|x - x'|_2, \tag{11}$$

which implies $\Delta \leq L|x - x'|_2$. Based on the Gaussian mechanism, when the noise scale satisfies

$$\sigma \geq \frac{\Delta \sqrt{2 \ln(1.25/\delta)}}{\epsilon}, \tag{12}$$

the perturbed feature $\tilde{h}_k$ satisfies (ϵ, δ)-differential privacy, i.e.,

$$\Pr[\tilde{h}(D) \in \mathcal{S}] \leq e^{\epsilon} \Pr[\tilde{h}(D') \in \mathcal{S}] + \delta, \tag{13}$$

for any adjacent datasets D and D' and any measurable set $\mathcal{S}$.

Furthermore, due to the zero-mean property of Gaussian noise, the expectation of the perturbed feature remains unchanged, i.e.,

$$\mathbb{E}[\tilde{h}_k] = \mathbb{E}[h_k + \mathcal{N}(0, \sigma^2 I)] = h_k, \tag{14}$$

indicating that the perturbation mechanism is unbiased. However, the variance of the perturbed feature is given by

$$\mathrm{Var}[\tilde{h}_k] = \mathrm{Var}[\mathcal{N}(0, \sigma^2 I)] = \sigma^2 I, \tag{15}$$

which introduces additional uncertainty into the feature representation and affects the stability of the training process.

To further analyze this effect, consider the gradient of the loss function $\mathcal{L}(\theta)$ with respect to the perturbed feature. Denoting $\epsilon_k \sim \mathcal{N}(0, \sigma^2 I)$, we have

$$\nabla \mathcal{L}(\tilde{h}_k) = \nabla \mathcal{L}(h_k + \epsilon_k). \tag{16}$$

Applying the first-order Taylor expansion yields

$$\nabla \mathcal{L}(\tilde{h}_k) \approx \nabla \mathcal{L}(h_k) + \nabla^2 \mathcal{L}(h_k)\epsilon_k, \tag{17}$$

from which it follows that the expectation of the gradient remains unchanged while its variance increases proportionally to σ^2. This implies that the injected noise does not introduce bias but may slow down convergence due to increased gradient fluctuations.

Consequently, the noise scale σ plays a critical role in balancing privacy preservation and model performance. A smaller σ leads to higher accuracy but weaker privacy protection, whereas a larger σ enhances privacy at the cost of degraded convergence behavior. This trade-off can be formally expressed as

$$\min_{\sigma} \ \mathcal{L}(\theta; \sigma) \text{s.t.} (\epsilon, \delta)\text{-DP}, \tag{18}$$

where $\mathcal{L}(\theta; \sigma)$ denotes the loss under noise level σ.

Privacy Cost Accumulation:In practical training scenarios, the learning process is conducted over multiple communication rounds. According to the composition property of differential privacy, the privacy loss accumulates over iterations.

Specifically, if each training round satisfies (ϵ, δ)-differential privacy, then after T rounds, the overall privacy guarantee can be bounded by $(T\epsilon, T\delta)$ under basic composition.

This implies that as the number of training rounds increases, the effective privacy protection becomes weaker. Therefore, there exists an inherent trade-off between training performance and privacy preservation. A larger number of iterations improves model accuracy but leads to higher privacy cost, while stronger noise injection reduces privacy loss at the expense of convergence speed.

Consequently, the noise scale and the number of training rounds should be carefully chosen to balance privacy budget consumption and model utility. In summary, by introducing Gaussian perturbation at the intermediate feature level, the proposed mechanism provides formal differential privacy guarantees while preserving unbiased feature representations. The interplay between noise intensity and training stability enables a flexible trade-off between privacy and utility, making the approach well suited for distributed learning in heterogeneous environments.

4. Theoretical Analysis

4.1. USFL Algorithm

Based on the aforementioned model splitting strategy and privacy-preserving mechanism, the U-shaped split federated learning process can be formulated as an iterative optimization algorithm. The global model parameter θ is updated through collaborative interactions between clients and edge servers under distributed settings.

Let $\theta^{(t)}$ denote the global model parameter at iteration t. For each client $k \in \mathcal{K}$, the local update can be represented as a composite operator

$$\theta_k^{(t+1)} = \mathcal{U}_k(\theta^{(t)}; \mathcal{D}_k, \sigma), \tag{19}$$

where $\mathcal{U}_k(\cdot)$ denotes the local update process jointly executed by the client and the edge server under the U-shaped split learning structure, and σ represents the noise intensity introduced by the privacy mechanism.

This local update can be interpreted as a stochastic gradient descent step over perturbed feature representations, given by

$$\theta_k^{(t+1)} = \theta^{(t)} - \eta \nabla \mathcal{L}_k(\tilde{h}_k; \theta^{(t)}), \tag{20}$$

where η is the learning rate and $\tilde{h}_k$ denotes the noise-perturbed intermediate feature. Since the privacy mechanism operates only on the feature level, the update can be regarded as gradient descent under stochastic input perturbations.

After all clients complete their local updates, a federated aggregation operator is applied to obtain the global model for the next iteration, i.e.,

$$\theta^{(t+1)} = \sum_{k=1,j=1}^{K} \frac{|D_k|}{|D_j|} \theta_k^{(t+1)} \tag{21}$$

Therefore, the overall U-shaped split federated learning algorithm can be expressed in a unified iterative form as

$$\theta^{(t+1)} = \mathcal{A}\Big(\{ U_k(\theta^{(t)}; D_k, \sigma) \}_{k=1}^{K} \Big). \tag{22}$$

This formulation reveals the essential structure of the proposed method: distributed optimization is achieved through the combination of local stochastic updates under feature perturbation and global aggregation across clients. Compared with conventional federated learning, the proposed approach reduces computational burden on resource-constrained clients via model splitting, while enhancing privacy protection through feature-level perturbation.

The above unified representation provides a foundation for subsequent convergence and complexity analysis.

4.2. Convergence and Complexity Analysis

Based on the unified algorithmic formulation in Section 4.1, we further analyze the proposed U-shaped split federated learning method from the perspectives of convergence and computational complexity.

We first investigate the convergence behavior of the model. Let the global objective function $F(\theta)$ be L-smooth, i.e.,

$$|\nabla F(\theta_1) - \nabla F(\theta_2)| \leq L|\theta_1 - \theta_2|. \tag{23}$$

Assume further that the stochastic gradient estimator remains unbiased under feature perturbation, namely

$$\mathbb{E}[\nabla \mathcal{L}_k(\tilde{h}_k; \theta)] = \nabla F(\theta), \tag{24}$$

which holds due to the zero-mean property of the injected Gaussian noise.

We then characterize the effect of privacy perturbation on gradient variance. Specifically, the perturbed gradient satisfies

$$\mathbb{E}\left[|\nabla \mathcal{L}_k(\tilde{h}_k; \theta) - \nabla F(\theta)|^2 \right] \leq \sigma_g^2 + c\sigma^2, \tag{25}$$

where σ_g^2 denotes the intrinsic variance of stochastic gradients, σ^2 is the noise variance introduced by the privacy mechanism, and c is a constant dependent on the model structure. This relation indicates that privacy perturbation increases the variance of gradient estimation, thereby affecting convergence behavior.

Under the above assumptions, applying standard stochastic optimization analysis yields the following convergence bound:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[|\nabla F(\theta^{(t)})|^2\right] \leq \mathcal{O}\left(\frac{F(\theta^{(0)})-F^*}{\eta T}\right) * \mathcal{O}(\eta(\sigma_g^2 + c\sigma^2)), \tag{26}$$

where F^* denotes the optimal objective value. This result shows that, with an appropriate choice of learning rate η, the algorithm converges to a neighborhood of the optimal solution, whose size depends on the noise level σ. In particular, larger noise leads to slower convergence and a larger steady-state error.

We further analyze the computational complexity of the proposed USFL framework. Let $C^{(1)}$, $C^{(2)}$, and $C^{(3)}$ denote the computational costs of the front-end, intermediate, and back-end sub-models, respectively.

In conventional federated learning, each client performs full model training with complexity $\mathcal{O}(C^{(1)} + C^{(2)} + C^{(3)})$. In contrast, under the proposed USFL framework, the client-side complexity is reduced to $\mathcal{O}(C^{(1)} + C^{(3)})$, while the intermediate computation $\mathcal{O}(C^{(2)})$ is offloaded to the edge server.

Therefore, the proposed method significantly reduces the computational burden on resource-constrained clients. The overall system complexity remains comparable to that of centralized training, but with distributed execution across clients and servers.

From the communication perspective, each training round requires the transmission of intermediate features and gradients between clients and servers. The corresponding communication cost can be expressed as $\mathcal{O}(S_h + S_g)$, where S_h and S_g denote the sizes of intermediate features and gradients, respectively. Compared with conventional federated learning that transmits full model parameters, the proposed approach can significantly reduce communication overhead, especially for deep models.

In summary, the proposed U-shaped split federated learning method achieves a favorable trade-off among convergence performance, computational efficiency, and communication cost. While the introduction of privacy noise increases gradient variance and slightly degrades convergence speed, it provides formal privacy guarantees and enables secure distributed learning in heterogeneous environments.

5. Experimental Results and Analysis

5.1. Experimental Setup

To evaluate the effectiveness of the proposed method, experiments are conducted under a space–air–ground–sea integrated network scenario using a real-world maritime dataset. Specifically, the SeaShips dataset is adopted, which contains various types of vessel images, including container ships, bulk carriers, and oil tankers [30]. The dataset is characterized by complex backgrounds and multi-scale objects, making it suitable for evaluating model performance in realistic maritime environments. The SeaShips dataset is selected as a representative maritime scenario in space–air–ground–sea integrated networks. Although the experiments are conducted on maritime data, the proposed framework is general and can be extended to other application domains.

To simulate heterogeneous conditions in space–air–ground–sea integrated networks, we assign different computational capabilities and communication rates to different types of nodes. Specifically, satellite nodes are modeled as resource-limited and high-latency devices with lower processing speeds and reduced communication rates. In contrast, maritime and ground nodes are configured with relatively higher computational capabilities and more stable communication links.

This heterogeneous configuration allows us to emulate realistic scenarios where different nodes exhibit significant disparities in computation and communication, and enables evaluation of the proposed method under such conditions.

For the model architecture, we employ ResNet18 as the backbone network due to its balanced trade-off between representational capability and computational efficiency. Under the U-shaped split learning framework, ResNet18 is partitioned into three segments: the shallow convolutional layers are deployed at the client side for initial feature extraction, the intermediate residual blocks are offloaded to the edge server for computation-intensive processing, and the final fully connected layers are placed back at the client side to complete prediction and loss evaluation. This configuration forms a "client–server–client" execution path.

To comprehensively assess the proposed approach, we consider several representative baselines, including centralized learning, federated learning (FL), split learning (SL), and split federated learning (SFL). In addition, we evaluate the proposed U-shaped split federated learning (USFL) method and its privacy-enhanced variant (USFL with DP), in order to examine the combined effects of model splitting and differential privacy. All methods are trained under identical hyperparameter settings to ensure a fair comparison, as summarized in Table 1.

Table 1. Experimental hyperparameter settings

Parameter	Value
Learning rate (η)	0.01
Optimizer	SGD
Momentum coefficient	0.9
Batch size	32
Training rounds (T)	100
Number of clients	5
Model architecture	ResNet18
Noise mechanism	Gaussian noise
Noise scale (σ)	0.1, 0.5, 1.0, 2.0
Model partition	Client–Server–Client (U-shaped)

Furthermore, to simulate resource-constrained and heterogeneous environments in SAGSIN, we introduce variability in computational capabilities and communication bandwidth across clients and edge nodes. This setting reflects practical deployment scenarios where devices exhibit diverse resource conditions. All experiments are repeated multiple times, and the average results are reported to ensure statistical reliability.

5.2. Performance Comparison

From a privacy perspective, centralized learning and distributed learning differ in several key aspects. In centralized learning, raw data and labels are directly uploaded to a central server, resulting in full data exposure. In contrast, in the proposed distributed framework, both data and labels remain local at client devices, and only intermediate features and gradients are transmitted.

Moreover, the proposed method incorporates a differential privacy mechanism, where the privacy level is controlled by the noise parameter σ. This introduces an additional privacy control dimension that is absent in centralized learning. Therefore, the comparison between the two paradigms is reflected in differences in data access, transmitted information, and privacy control mechanisms.

To evaluate the performance of the proposed method, we compare it with several representative approaches, including centralized learning, FL, SL, and SFL. Figures 4 and 5 illustrate the accuracy curves of different methods on the training and test sets, respectively, as functions of the number of training rounds.

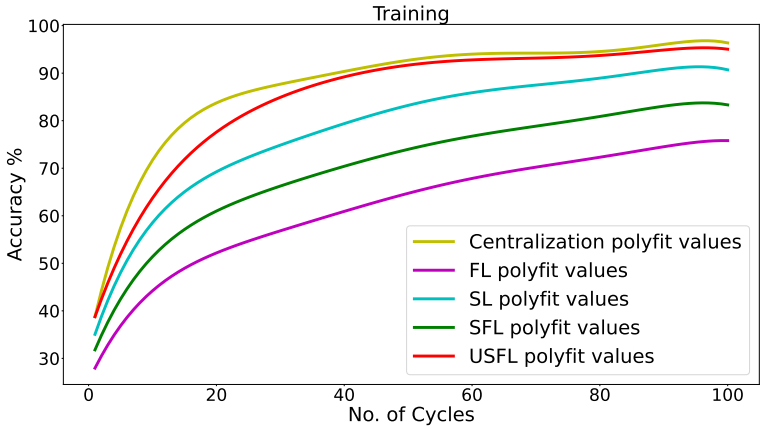


Figure 4. Training accuracy comparison of different learning methods over communication rounds.

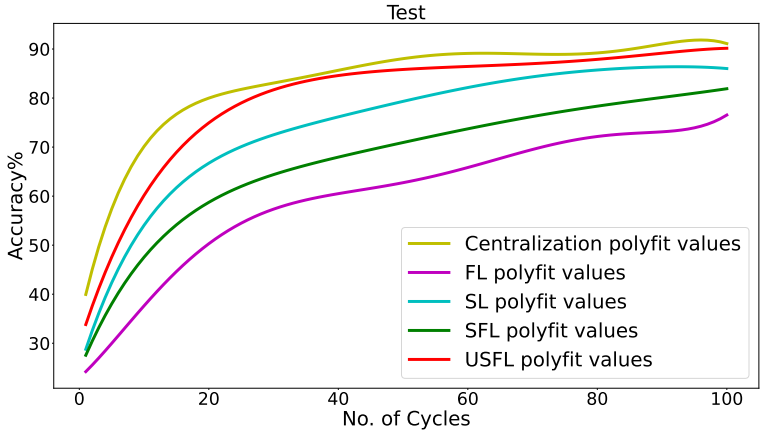


Figure 5. Test accuracy comparison of different learning methods over communication rounds.

From the training results, all methods exhibit a consistent upward trend as the number of training rounds increases; however, significant differences can be observed in terms of convergence speed and final accuracy. Centralized learning achieves the best performance, reaching over 96% training accuracy and 91.98% test accuracy, as it has access to the complete dataset. In contrast, the traditional FL method suffers from slow convergence and significantly lower final accuracy (75.64% test accuracy), mainly due to data heterogeneity and limited information exchange among clients.

The SL method improves training efficiency by offloading part of the model to the server, leading to noticeable performance gains over FL, achieving 86.85% test accuracy. However, its performance is still limited by the absence of global parameter aggregation. The SFL method further enhances convergence stability compared with SL, but its test accuracy (82.43%) remains lower than more advanced collaborative frameworks.

The proposed USFL method consistently outperforms FL, SL, and SFL throughout the training process. As shown in Figures 4 and 5, the USFL curve closely follows that of centralized learning and rapidly reaches high accuracy within fewer training rounds. Specifically, USFL achieves 90.23% test accuracy, which is significantly higher than FL, SL, and SFL, and approaches the performance of centralized learning.

In addition, as summarized in Table 2, USFL achieves a runtime of 336.47 min, which is lower than FL (484.72 min) and SFL (360.28 min), and comparable to centralized learning (338.91 min). This indicates that the proposed method effectively reduces computational overhead while maintaining high performance.

Table 2. Performance comparison of different learning methods.

Method	Runtime (min)	Train Accuracy (%)	Test Accuracy (%)
Centralized Learning	338.91	96.10	91.98
FL	484.72	76.25	75.64
SL	317.77	90.84	86.85
SFL	360.28	85.25	82.43
USFL (Proposed)	**336.47**	**92.39**	**90.23**

The test results exhibit trends consistent with those observed during training. USFL demonstrates strong generalization capability, achieving performance close to centralized learning while significantly outperforming FL, SL, and SFL.

Overall, the results demonstrate that the proposed USFL method achieves a better trade-off among accuracy, convergence speed, and computational efficiency. It effectively bridges the performance gap between centralized and distributed learning, providing an efficient and privacy-preserving solution for integrated network environments.

5.3. Impact of Differential Privacy

To evaluate the impact of the differential privacy mechanism on model performance, we conduct experiments under the USFL framework with different noise scales σ. The training and test accuracies under varying noise levels are illustrated in Figure 6 and Figure 7, respectively.

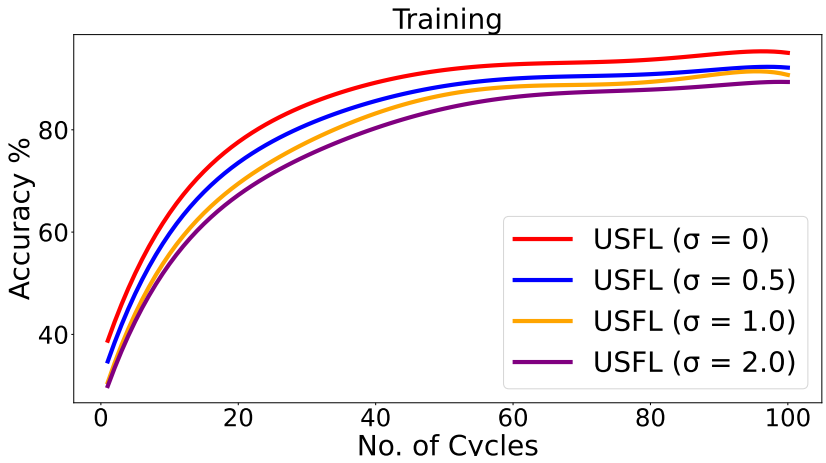


Figure 6. Training accuracy under different noise levels.

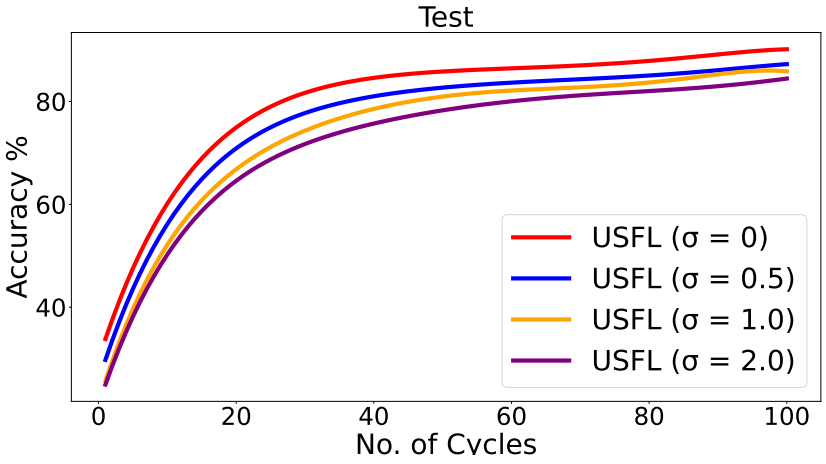


Figure 7. Test accuracy under different noise levels.

From the training results, it can be observed that when $\sigma = 0$, i.e., no privacy perturbation is applied, the model achieves the fastest convergence rate and the highest final accuracy. As the noise scale increases (e.g., $\sigma = 0.5, 1.0, 2.0$), the convergence speed gradually decreases, and the final accuracy is correspondingly reduced. This behavior indicates that the introduction of privacy noise affects the training dynamics, primarily due to the increased uncertainty in gradient estimation.

The test results exhibit a similar trend. A small noise level (e.g., $\sigma = 0.5$) only causes a minor degradation in performance, whereas larger noise levels (e.g., $\sigma = 2.0$) lead to a more noticeable decline in generalization accuracy. This suggests that an appropriate choice of the noise parameter can effectively balance privacy preservation and model performance.

These observations are consistent with the theoretical analysis presented in Sections 3 and 4. Specifically, the injected noise increases the variance of intermediate feature representations, which propagates to the gradient updates and results in slower convergence. This phenomenon aligns with the derived convergence bound, where the noise variance directly influences the optimization error.

Furthermore, it can be observed that the performance gap between different noise levels is more pronounced in the early training stages, while the curves gradually stabilize as training progresses. This indicates that although privacy perturbation introduces additional noise into the learning process, the model is still capable of learning meaningful representations over time and eventually reaching a stable performance level.

In summary, the differential privacy mechanism introduces a controllable trade-off between privacy protection and learning performance. While stronger privacy guarantees (larger σ) lead to reduced accuracy and slower convergence, moderate noise levels can preserve most of the model performance while providing effective privacy protection. These results validate the theoretical privacy–utility trade-off and provide practical guidance for selecting appropriate noise parameters in real-world deployments.

5.4. Communication and Statistical Analysis

To quantitatively characterize the statistical stability of the proposed USFL framework in the space–air–ground–sea integrated network, the coefficient of variation (CV) is adopted as a key metric to measure the dispersion of model performance. The CV is defined as the ratio of the standard deviation to the mean,

$$\mathrm{CV} = \frac{\sigma}{\mu}, \tag{27}$$

where σ denotes the standard deviation of the model accuracy and μ represents the corresponding mean value over communication rounds. As a dimensionless quantity, the CV effectively captures the fluctuation behavior and convergence stability of the learning process.

As illustrated in Figures 8 and 9, the evolution of the CV over 200 communication rounds is analyzed for the proposed USFL method. From the simulation results, it can be observed that during the training phase, the CV values remain within a relatively low range and gradually decrease as the number of communication rounds increases. This indicates that the training process exhibits strong convergence consistency. Although slight fluctuations appear in the early stages, the CV stabilizes as the global model is iteratively updated, demonstrating a clear convergence trend.

In the test phase, the CV values are slightly higher compared to those in the training phase. This can be attributed to the distribution discrepancy between training and testing datasets. Nevertheless, the CV remains within a controlled and stable range without significant oscillations or divergence, indicating that the model maintains robust generalization performance.

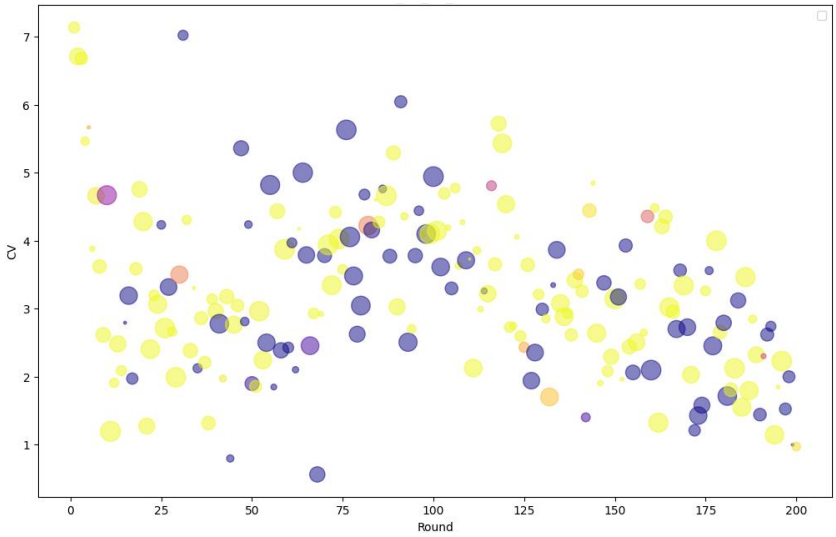


Figure 8. The CV of model accuracy during training rounds.

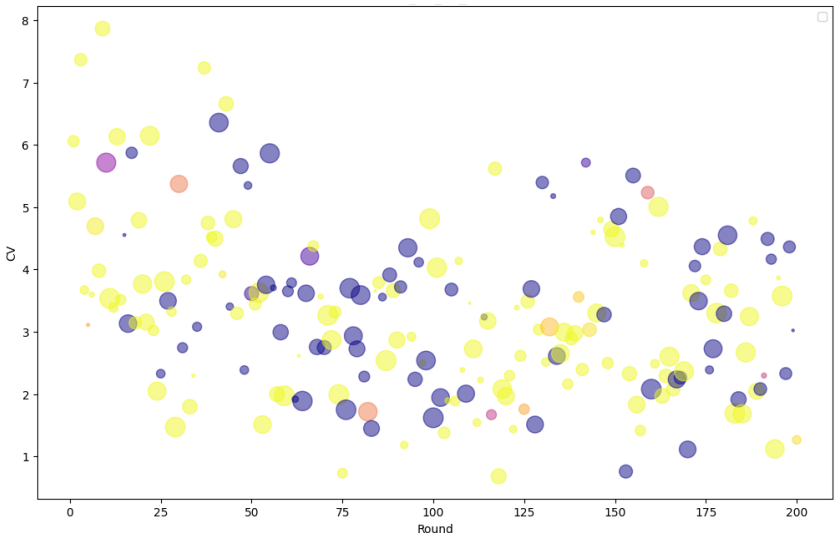


Figure 9. The CV of model accuracy during test rounds.

Further analysis reveals that the proposed USFL framework exhibits favorable statistical consistency in distributed learning environments. This property mainly stems from the U-shaped architecture, where raw data remain localized at the client side and only intermediate representations and gradients are transmitted. Such a mechanism effectively reduces the impact of communication noise and system uncertainty on the training process. Moreover, the collaborative update among multiple clients and edge servers helps smooth local variations, thereby enhancing the overall stability of the global model.

As illustrated by Figures 8 and 9, the values remain concentrated within a bounded interval and do not exceed commonly accepted stability thresholds. This suggests that the model outputs exhibit a stable statistical distribution throughout the training and inference processes. Therefore, the proposed USFL method not only achieves efficient learning but also ensures reliable statistical behavior under complex and resource-constrained environments.

In summary, the CV-based analysis validates the stability and robustness of the USFL framework in distributed collaborative learning. These results provide both theoretical and empirical support for deploying USFL in practical space–air–ground–sea integrated network scenarios.

Comparative Analysis: Based on the above experimental results, we further provide a comparative analysis of different learning methods.

From an accuracy perspective, the proposed USFL method achieves performance close to centralized learning, indicating comparable learning capability. Compared with FL, SL, and SFL, the proposed method demonstrates faster convergence and improved stability, as reflected by smoother accuracy curves and lower CV values.

From a communication perspective, the proposed method reduces communication overhead by transmitting intermediate features instead of full model parameters, which is more efficient in resource-constrained environments.

From a privacy perspective, centralized learning exposes raw data, while the proposed method preserves data locality and incorporates differential privacy. The experimental results under different noise levels further illustrate the trade-off between privacy protection and model performance.

Overall, the proposed method achieves a better balance among accuracy, communication efficiency, and privacy preservation compared with existing approaches.

6. Conclusions

To address key challenges in space–air–ground–sea integrated networks, including resource constraints, data heterogeneity, and privacy preservation requirements, this paper proposes a distributed collaborative learning framework based on USFL method. By integrating the advantages of FL and SL, the proposed method achieves efficient model training and inference while preserving data privacy. At the system level, a distributed computing architecture tailored for the SAGSIN is developed, where clients, edge servers, and a parameter server are effectively coordinated to optimize both computational and communication resources. At the model level, a U-shaped structure is employed to partition the deep neural network, ensuring that raw data remain localized at the client side while only intermediate features and gradients are transmitted, thereby significantly reducing the risk of privacy leakage. At the methodological level, a differential privacy mechanism is incorporated into the intermediate parameter transmission process, providing privacy guarantees while maintaining model performance. From a theoretical perspective, the proposed method is analyzed from both optimization and statistical viewpoints. The impact of privacy noise on model convergence is characterized, and the underlying source of stability in distributed environments is explained from a mathematical standpoint. In the experimental evaluation, extensive simulations are conducted using the SeaShips dataset and the ResNet18 model. The results demonstrate that the proposed USFL method outperforms conventional approaches in terms of convergence speed, model accuracy, and stability. Furthermore, the analysis based on the CV confirms the statistical consistency and robustness of the model during both training and testing phases. In summary, the proposed USFL framework enables efficient, secure, and stable distributed learning in complex SAGSIN environments, showing strong potential for practical applications. Future work will focus on optimizing model partitioning strategies and developing adaptive mechanisms for differential privacy parameters, with the goal of further improving system performance and stability in dynamic and heterogeneous network scenarios.

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

Funding: This work was supported in part by the Key Basic Research Project of the Fundamental Reinforcement Program under Grant 86220020, the project "Research on Key Technologies of Multi-level Heterogeneous Networking and Resource Optimization Scheduling in the Sea Area" under Grant 83117038J, and by the project "Theory and Key Technologies of Intelligent Sensing, Communication, and Control for UAVs in Low-Altitude Security Scenarios" under Grant U25A20395.

Data Availability Statement: The datasets generated and/or analysed during the current study are not publicly available as the related research is still ongoing and the data are not yet suitable for public release, but are available from the corresponding author on reasonable request.

Acknowledgments: The authors would like to thank all individuals and organizations that provided support and assistance for this research.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Jin, D.; Kannengiesser, N.; Rank, S.; Sunyaev, A. Collaborative distributed machine learning. *Acm Comput. Surv.* **2024**, *57*, 1–36. [CrossRef]
2. Li, T.; Sahu, A.K.; Talwalkar, A.; Smith, V. Federated learning: Challenges, methods, and future directions. *IEEE Signal Process. Mag.* **2020**, *37*, 50–60. [CrossRef]
3. 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]
4. Kopetz, H.; Steiner, W. Internet of things. In *Real-Time Systems: Design Principles for Distributed Embedded Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 325–341.
5. Xu, D.; Li, T.; Li, Y.; Su, X.; Tarkoma, S.; Jiang, T.; Crowcroft, J.; Hui, P. Edge intelligence: Empowering intelligence to the edge of network. *Proc. IEEE* **2021**, *109*, 1778–1837. [CrossRef]
6. Vashishtha, P.K.; Kumari, S. A Conditional Privacy-Preserving Short Signature Scheme for Industrial Internet of Things. *J. Reliab. Secur. Comput.* **2026**, *2*, 39–49. [CrossRef]
7. Sun, X.; Yang, T.; Gao, S.; Wang, K.; Li, X. Convergence of S3C empowered distributed cooperative optimization for multi-unmanned surface vehicles. *Phys. Commun.* **2022**, *51*, 101566. [CrossRef]
8. Bai, X.; Li, B.; Xu, X.; Xiao, Y. A review of current research and advances in unmanned surface vehicles. *J. Mar. Sci. Appl.* **2022**, *21*, 47–58. [CrossRef]
9. Er, M.J.; Ma, C.; Liu, T.; Gong, H. Intelligent motion control of unmanned surface vehicles: A critical review. *Ocean. Eng.* **2023**, *280*, 114562. [CrossRef]
10. Laghari, A.A.; Jumani, A.K.; Laghari, R.A.; Nawaz, H. Unmanned aerial vehicles: A review. *Cogn. Robot.* **2023**, *3*, 8–22. [CrossRef]
11. Ahmed, F.; Mohanta, J.C.; Keshari, A.; Yadav, P.S. Recent advances in unmanned aerial vehicles: A review. *Arab. J. Sci. Eng.* **2022**, *47*, 7963–7984. [CrossRef]
12. Miikkulainen, R.; Liang, J.; Meyerson, E.; Rawal, A.; Fink, D.; Francon, O.; Raju, B.; Shahrzad, H.; Navruzyan, A.; Duffy, N.; et al. Evolving deep neural networks. In *Artificial Intelligence in the Age of Neural Networks and Brain Computing*; Elsevier: Amsterdam, The Netherlands, 2024; pp. 269–287.
13. Gawlikowski, J.; Tassi, C.R.N.; Ali, M.; Lee, J.; Humt, M.; Feng, J.; Kruspe, A.; Triebel, R.; Jung, P.; Roscher, R.; et al. A survey of uncertainty in deep neural networks. *Artif. Intell. Rev.* **2023**, *56*, 1513–1589. [CrossRef]
14. Hu, Z.; Zhou, T.; Wu, B.; Chen, C.; Wang, Y. A review and experimental evaluation on split learning. *Future Internet* **2025**, *17*, 87. [CrossRef]
15. Wu, W.; Li, M.; Qu, K.; Zhou, C.; Shen, X.; Zhuang, W.; Li, X.; Shi, W. Split learning over wireless networks: Parallel design and resource management. *IEEE J. Sel. Areas Commun.* **2023**, *41*, 1051–1066. [CrossRef]
16. Wang, Z.; Tian, M.; Qiu, Y.; Zhao, Z.; Zhang, Y.; Liu, K. Federated Learning Privacy Protection via Training Randomness. *Icck Trans. Sensing, Commun. Control* **2025**, *2*, 226–237.
17. Thapa, C.; Arachchige, P.C.M.; Camtepe, S.; Sun, L. Splitfed: When federated learning meets split learning. In *Proceedings of AAAI Conference on Artificial Intelligence*; AAAI Publications: Washington, DC, USA, 2022; Volume 36, pp. 8485–8493.
18. Shen, J.; Cheng, N.; Wang, X.; Lyu, F.; Xu, W.; Liu, Z.; Aldubaikhy, K.; Shen, X. RingSFL: An adaptive split federated learning towards taming client heterogeneity. *IEEE Trans. Mob. Comput.* **2023**, *23*, 5462–5478. [CrossRef]
19. Liu, X.; Deng, Y.; Mahmoodi, T. Wireless distributed learning: A new hybrid split and federated learning approach. *IEEE Trans. Wirel. Commun.* **2022**, *22*, 2650–2665. [CrossRef]

20. Li, Q.; Diao, Y.; Chen, Q.; He, B. Federated learning on non-iid data silos: An experimental study. In *Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE)*; IEEE: New York, NY, USA, 2022; pp. 965–978.
21. Liang, X.; Xu, Y.; Lin, Y.e.; Zhang, C. Federated split learning via dynamic aggregation and homomorphic encryption on non-IID data. *J. Supercomput.* **2025**, *81*, 63. [CrossRef]
22. Xu, C.; Li, J.; Liu, Y.; Ling, Y.; Wen, M. Accelerating split federated learning over wireless communication networks. *IEEE Trans. Wirel. Commun.* **2023**, *23*, 5587–5599. [CrossRef]
23. Jonnalagadda, A.K.; Bura, C. Immune-Inspired AI: Adaptive Defense Models for Intelligent Edge Environments. *Icck Trans. Emerg. Top. Artif. Intell.* **2025**, *2*, 157–168.
24. Sun, X.; Yang, T.; Han, C.; Lee, G.C.; Sun, S. Subarea Collaborative Deep Q-Network with Online Learning: Advanced Multi-Agent Routing Strategies for Dynamic Maritime Networks. *IEEE Open J. Commun. Soc.* **2025**, *6*, 7200–7214. [CrossRef]
25. Chen, M.; Gündüz, D.; Huang, K.; Saad, W.; Bennis, M.; Feljan, A.V.; Poor, H.V. Distributed learning in wireless networks: Recent progress and future challenges. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3579–3605. [CrossRef]
26. Höhn, D.; Mumm, L.; Reitz, B.; Tsiroglou, C.; Hahn, A. Enabling Future Maritime Traffic Management: A Decentralized Architecture for Sharing Data in the Maritime Domain. *J. Mar. Sci. Eng.* **2025**, *13*, 732. [CrossRef]
27. Sun, X.; Yang, T.; Han, C.; Lee, G.C.; Sun, S. An Efficient and Privacy-Enhanced U-Shaped Split Federated Learning Method for Distributed Multi-Client Systems. In *Proceedings of the 2025 IEEE International Conference on Communications Workshops (ICC Workshops)*; IEEE: New York, NY, USA, 2025; pp. 1432–1437.
28. Samek, W.; Montavon, G.; Lapuschkin, S.; Anders, C.J.; Müller, K.R. Explaining deep neural networks and beyond: A review of methods and applications. *Proc. IEEE* **2021**, *109*, 247–278. [CrossRef]
29. Liu, H.; Brailsford, T.; Bull, L. Resnet18 performance: Impact of network depth and image resolution on image classification. In *Proceedings of the 2024 8th International Conference on Advances in Artificial Intelligence*; Association for Computing Machinery: New York, NY, USA, 2024; pp. 351–356.
30. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]

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.