Article

Efficient Asynchronous Federated Learning for AUV Swarm

Zezhao Meng 1, Zhi Li 1,*, Xiangwang Hou 2, Jun Du 2, Jianrui Chen 2 and Wei Wei 3

1 School of Mechano-Electronic Engineering, Xidian University, Xi'an 710071, China
2 Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
3 Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen 514231, China

* Correspondence: zhli@xidian.edu.cn

Abstract: The development of automatic underwater vehicles (AUVs) has brought about unprecedented profits and opportunities. In order to discover the hidden valuable data detected by an AUV swarm, it is necessary to aggregate the data detected by AUV swarm to generate a powerful machine learning model. Traditional centralized machine learning generates a large number of data exchanges and faces problems of enormous training data, large-scale models, and communication. In underwater environments, radio waves are strongly absorbed, and acoustic communication is the only feasible technology. Unlike electromagnetic wave communication on land, the bandwidth of underwater acoustic communication is extremely limited, with the transmission rate being only 1/105 of the electromagnetic wave. Therefore, traditional centralized machine learning cannot support underwater AUV swarm training. In recent years, federated learning could only interact with model parameters without interacting with data, which greatly reduced communication costs. Therefore, this paper introduces federated learning into the collaboration of an AUV swarm. In order to further reduce the constraints of underwater scarce communication resources on federated learning and alleviate the straggler effect, in this work, we designed an asynchronous federated learning method. Finally, we constructed the optimization problem of minimizing the weighted sum of delay and energy consumption, relying on jointly optimizing the AUV CPU frequency and signal transmission power. In order to solve this complex optimization problem of high-dimensional non-convex time series accumulation, we transformed the problem into a Markov decision process (MDP) and use the proximal policy optimization 2 (PPO2) algorithm to solve this problem. The simulation results demonstrate the effectiveness and superiority of our method.

Keywords: federated learning (FL); autonomous underwater vehicle (AUV); gradient compression; communication resource optimization; proximal policy optimization 2 (PPO2)

1. Introduction

An automatic underwater vehicle (AUV) is a kind of submarine robot that can carry out ocean sampling activities independently; it is widely used in underwater research and the marine industry. For example, AUVs have been used to sample coastal frontiers, monitor coastal areas, measure thermocline turbulence, obtain interdisciplinary data, conduct fishery research, and install submarine cables under the frozen sea [1,2]. However, due to the limited capacity of a single AUV, processing more complex tasks by AUV swarm has become a hot research topic. In recent years, underwater research based on the AUV group has become more popular [3,4]. The advantages of the AUV swarm cooperation over a single AUV can be summarized into two points. On the one hand, an AUV swarm can perform tasks that cannot be completed by a single AUV, such as underwater hydrological characteristic modeling, on the other hand, an AUV swarm can perform tasks that cannot be completed by a single AUV, such as rounding-up targets, formation cruising, collaborative positioning, etc. In the above two aspects, it is often critical to establish a powerful machine learning model to mine the values behind the observed data, or to improve the effectiveness of the AUV’s own actions. Specifically, for the sake of exploring the hidden valuable information of the
data detected by the AUV swarm, it is necessary to aggregate the data detected by the AUV swarm to generate a powerful machine learning model, such as forming the hydrological characteristics model of the water area by data fitting. Moreover, in order to improve the movement of the AUV, it is a requisite to train the AUV to obtain a strong reinforcement learning model to achieve efficient autonomous action of the AUV. In order to support the above two goals, traditional centralized machine learning needs to interact with a large number of data, which will require a lot of communication resources. However, unlike on land, underwater electromagnetic wave communications with high bandwidths cannot be used; they often rely on underwater acoustic communication for data transmission. The bandwidth of underwater acoustic communication is very limited, and the transmission rate is only \(1/10^5\) of the electromagnetic wave. Therefore, traditional centralized machine learning cannot support underwater AUV swarm training and the production of large models. Federated learning, a new machine learning framework, was first proposed by Google in 2016 [5]. Each data holder uses his/her own data to train a model, and the models interact with each other. Finally, a global model is obtained through model aggregation. Unlike traditional centralized machine learning, federated learning can only interact model parameters without interacting with data, which can greatly reduce communication overhead and reduce dependence on communication resources, and is suitable for the environment where underwater communication resources are scarce. Therefore, this paper introduces federated learning into the collaboration of the AUV swarm. Furthermore, considering that the interaction model parameters are much less than the interactions with the original data, there is still a large number of data, and the traditional federated learning method (i.e., FEDAVG proposed by Google [5]) is synchronized, which is prone to the emergence of the straggler effect, especially in unreliable underwater information transmission and large time delays. Therefore, this paper designs an asynchronous federated learning method to further reduce data transmission and reduce the straggler effect, which will play an important role in the harsh underwater environment. Furthermore, considering the limited energy of the AUV and the time requirement of model training, we constructed an optimization problem of minimizing the weighted sum of delay and energy consumption, relying on jointly optimizing AUV CPU frequency and signal transmission power. In order to solve this complex optimization problem of high-dimensional non-convex time series accumulation, we built the problem into an MDP and used the PPO2 algorithm to solve this problem. The main contributions are as follows:

- To the authors’ knowledge, this paper is the first to introduce federated learning into the AUV swarm. Federated learning can help AUV swarm train large-scale machine learning models in an environment where underwater communication resources are scarce.
- In order to further reduce the constraints of underwater scarce communication resources on federated learning, we designed an asynchronous federated learning method, which can effectively alleviate the straggler effect and further reduce data interaction.
- We constructed the optimization problem of minimum delay and energy consumption, by jointly optimizing AUV CPU frequency and signal transmission power.
- In order to solve the optimization problem efficiently, we converted it into an MDP and proposed the PPO2 algorithm to solve this problem. The simulation results verify the effectiveness of the proposed algorithm.

The content and organizational structure of this article are as follows:

- In the third section, a federated learning model based on the AUV swarm is established. First, we established a federal learning model. Second, the communication system model was established. Then the control model was established, the nodes participating in this round of upload were selected locally, and redundant nodes were adaptively skipped. Then a delay model was established to calculate the time and
total time of each phase of federated learning. Then the energy consumption model was established as the energy constraint. Finally, the optimization problem is listed.

- In the fourth section, the PPO2 algorithm is used to solve the optimization problem.
- The fifth section is the experimental part. By changing the relevant parameters, we can observe the communication times, model accuracy, and model The simulation results show the convergence of the PPO2 algorithm.
- The last section of this paper is the summary, which summarizes the main points and shortcomings of this work.

2. Related Work

The bottleneck limiting the performance of the AUV swarm mainly lies in the difficulties of underwater communications [6]. When AUV groups are used for federated learning tasks, as the scale of task models and AUV groups expand, the pressure on the self-organizing data interaction network increases, which requires designing algorithms to save communication resources as much as possible. However, reducing communication resources excessively may lead to compromised federated learning performance and confusion about the AUV formations, which creates a paradox. How to make a trade-off between resource consumption and model performance is worth studying at present.

AUV swarm has the advantages of rapid deployment, controllable action, flexible networks, and other advantages because of its formation, dynamic, mobility, and other characteristics. In recent years, the research into underwater exploration systems based on multiple AUVs or AUV groups has become popular, especially in navigation or path planning [7], collaborative data collection [8], target hunting [9], and so on. For example, in the offshore oil and gas industry, AUVs equipped with underwater environment monitoring sensors cooperate to effectively detect the boundary of oil and gas producing areas. Cui et al. [10] proposed an adaptive path planning algorithm based on the random tree star algorithm to estimate the scalar field underwater for the AUV group. Noguchi et al. [11] put forward a dynamic task allocation and path planning algorithm for the AUV group, which gives full play to the advantages of the AUV group. Huang et al. [12] used the AUV group equipped with scanning imaging sonar, and proposed using the binary Bayesian filter to process the input signal. Then, a path-planning strategy based on the artificial potential field was proposed for real-time target tracking. Cao et al. [13] used the artificial potential field theory to model the underwater environment information and introduce the potential field function into the dispersion, similarity, and difference of the potential field function. On this basis, an intelligent path planning model that can achieve accurate AUV swarm collaborative target search is proposed.

In the federated learning aggregation algorithm, each participant needs to completely send local parameters to the server for model aggregation. These model parameters tend to have a lot of data, which brings huge communication overhead. In [14], the author protects the privacy of the federated learning architecture through covert communication technology, minimizes the federated learning delay by optimizing the interference power, signal power, and training accuracy, and uses an alternating descent algorithm to solve the optimization problem. In [15], the author uses federated learning to build a digital twin model based on the digital twin edge network architecture, uses an asynchronous model update scheme, and performs device selection locally, minimizing channel allocation, and CPU frequency as variables. The federated learning delay uses the artificial intelligence method DNN (deep neural network) model to dynamically solve the optimization problem. In [16], the author selects some clients to send their local model parameters to the server, avoids the training loss, and convergence time being affected by a reasonable local selection strategy, and establishes an optimization problem to optimize the convergence time and training loss, using ANN (artificial neural network) to solve optimization problems. In [17], the author defines a new performance evaluation criterion, i.e., learning efficiency, which is discussed in scenarios where the device is equipped with CPU and GPU, respectively, e.g., channel resource allocation and bandwidth allocation as optimization variables were
used to maximize learning efficiency via the Lagrange multiplier method, KKT (Karush-Kuhn-Tucker) conditions. A two-dimensional search algorithm was used to solve the optimization problem and detect and skip some redundant communication rounds locally with adaptive communication rules; a comparable convergence speed was used to the original SGD.

In [18], the authors proposed a communication-saving method for distributed machine learning, which is a gradient-based selection method that reduces redundant communication rounds. In [19], the author introduced the above method [18] into federated learning, based on the traditional SGD (stochastic gradient descent) method. In [20], the author delves into the basis of the above work, and used the gradient sparse method to select redundant nodes locally to reduce the communication amount. The work of this paper continues on this basis. So far, there is no work to introduce federal learning into AUV swarm.

3. System Model

This paper considers the following scenario: a fixed-form swarm consisting of a leader AUV $L$ and a set $M$ of the $M$ follower AUV sailing at the same depths with constant speed, collecting seabed data, cooperating, and using federated learning to perform various machine learning tasks, such as target recognition, path planning, etc. Each AUV maintains the same distance from other AUVs and navigators, collecting data with their own independent sensors, and training local models. The local model parameters are then sent to the navigator using the uplink. The leader AUV uses the information uploaded by the follower to aggregate into global model parameters and sends it to the follower using the downlink. The follower proceeds with a new round of local training using the received global model parameters. In order to keep AUVs in formation, AUV $L$ and AUV $m$ also need to transfer position information and speed information to each other. In each round, AUV $m$ uploads its current speed and position data to AUV $L$. AUV $L$ calculates and broadcasts the subsequent speed and direction from the received information.

3.1. Federated Learning Model

Assuming that the input data collected by the follower AUV $m \in M$ is $X_m$, the output data are $Y_m$, and the local model parameter is $w_m$, then the data set of the AUV $m$ can be expressed as

$D_m = \{(x_{m,1}, y_{m,1}), (x_{m,2}, y_{m,2}), \ldots, (x_{m,N_m}, y_{m,N_m})\}$, where $N_m$ is the number of the data that AUV $m$ owns, the loss function on its data set $D_m$ can be expressed as

$F_m(w_m) = \frac{1}{N_m} \sum_{i=1}^{N_m} f(w_m; x_{m,i}, y_{m,i}), \quad \forall m \in M \tag{1}$

Assuming that the global model parameter is $w$, for an AUV $m \in M$ containing $N_m$ data, the corresponding local loss function is weighted, averaged, and defined as the global loss function:

$F(w) \triangleq \sum_{m=1}^{M} \frac{N_m F_m(w)}{N} = \frac{1}{N} \sum_{m=1}^{M} \sum_{i=1}^{N_m} f(w_m; x_{m,i}, y_{m,i}) \tag{2}$

The purpose of federated learning is to find a parametric model that minimizes the global loss function, i.e., the optimal parametric model can be expressed as

$w^* = \arg \min F(w)$, \quad \tag{3}$

where $w = w_1 = \cdots = w_M$.

3.2. Communication Model

In the underwater environment, electromagnetic waves and optical signals attenuate quickly and travel over short distances. Therefore, the most commonly used underwater
communication is underwater acoustic communication. The attenuation of the water signal with frequency $f$ at distance $D$ can be given by

$$A(D, f) = k^2 a(f)^D$$ (4)

where $k$ represents the spreading factor, and $a(f)$ is the absorption coefficient, which can be expressed by the following empirical formula [21]:

$$10 \log a(f) = 0.11 f^2 + \frac{44 f^2}{4100 + f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003.$$ (5)

In the marine environment, noise is mainly divided into turbulence noise, shipping noise, wave noise, and thermal noise, respectively. According to [22], the power spectral density (p.s.d) of four main types of noise in dB re $\mu$Pa per Hz on the communication frequency $f$ can be given by:

$$10 \log N_\theta(f) = 17 - 30 \log f$$ (6)

$$10 \log N_s(f) = 40 + 20 \left(s - \frac{1}{2}\right) + 26 \log f - 60 \log(f + 0.03)$$ (7)

$$10 \log N_w(f) = 50 + 7.5w^{\frac{1}{2}} - 40 \log f - 40 \log(f + 0.4)$$ (8)

$$10 \log N_{th}(f) = -15 + 20 \log f$$ (9)

Moreover, $s \in [0, 1]$ is the shipping activity factor, while $w$ represents the wind velocity (m/s). Hence, the combined noise $N(f)$ can be represented as

$$N(f) = N_\theta(f) + N_s(f) + N_w(f) + N_{th}(f)$$ (10)

Therefore, the normalized SNR of a signal with unit transmitted power and bandwidth can be represented as

$$\gamma(D, f) = \frac{1}{A(D, f) N(f)}.$$ (11)

The uplink data rate between AUV $m$ and leader AUV $L$ is given by

$$R_{m}^U = B_{m}^U \log_2\left(1 + \frac{p_m \gamma(D, f)}{B_{m}^U}\right),$$ (12)

where $B_{m}^U$ is the allocated uplink bandwidth of the AUV $m$, while $p_m \in (0, p_{max})$ is the transmitting power of the AUV $m$.

In the iteration, the leader AUV broadcasts the information to the followers and, hence, the downlink data rate between AUV $m$ and leader AUV $L$ can be represented as

$$R_{m}^D = B_{m}^D \log_2\left(1 + \frac{p_L \gamma(D, f)}{B_{m}^D}\right)$$ (13)

where $B_{m}^D$ is the downlink bandwidth, $p_L \in (0, p_{max})$ denotes the transmitting power of leader AUV.

### 3.3. Control Model

Due to the high cost of underwater communication, most gradient interactions in the distributed SGD are redundant. Therefore, this paper introduces the concept of lazy nodes, allowing each follower drone to perform self-detection locally so that some nodes skip some rounds of communication. The definition of a lazy node in this paper refers to the work of Chen et al. [19].
\[
\frac{\|\nabla_{\mathcal{M}_N}^{t-1}\|^2}{M_N} \leq \frac{\|\nabla_{\mathcal{M}}^{t-1}\|^2}{M}
\]  
(14)

Among them, these lazy nodes constitute a set \(\mathcal{M}_N\) of size \(M_N\). In this paper, the gradient descent algorithm is used to optimize the global model, where \(\gamma\) represents the learning rate

\[
w(t) = w(t-1) - \gamma \nabla_{\mathcal{M}}^{t-1}
\]
(15)

\[
\frac{\|\nabla_{\mathcal{M}}^{t-1}\|^2}{M} \leq \frac{M_N}{\gamma^2 M} \|w(t) - w(t-1)\|^2,
\]
(16)

Since the global model tends to converge, the following approximation is used

\[
w(t) - w(t-1) \approx w(t-1) - w(t-2)
\]
(17)

According to the mean inequality, we have

\[
\left\| \nabla_{\mathcal{M}_N}^{t-1} \right\|^2 = \left\| \sum_{m \in \mathcal{M}_N} \frac{N_m \nabla_{\mathcal{M}}^{t-1}}{N} \right\|^2 \leq \frac{M_N}{N^2} \sum_{m \in \mathcal{M}_N} \left\| N_m \nabla_{\mathcal{M}}^{t-1} \right\|^2
\]
(18)

Let \(M_N = \beta M\), Equation (14) can be deduced as Equation (18)

\[
\left\| N_m \nabla_{\mathcal{M}}^{t-1} \right\|^2 \leq \frac{N^2}{\gamma^2 M^2 \beta^2} \|w(t-1) - w(t-2)\|^2.
\]
(19)

At one round \(t\), AUV \(m\) checks locally whether Equation (18) is satisfied. If satisfied, skip this round of uploading, otherwise participate in this round of uploading. Considering the extreme case, if all nodes in a certain round \(t\) satisfy Equation (18), then AUV \(L\) cannot receive the model information uploaded by the follower AUV. At this time, AUV \(L\) randomly selects some follower drones to participate in the upload after a certain time interval. In this paper, we randomly select 1 AUV \(m\) to participate in uploading in extreme cases.

In order to prevent AUVs from not communicating for a long time due to the setting of lazy nodes at certain times, we require that AUVs communicate at least once within \(\tau\) round. Each AUV locally records whether it uploads itself in each round of uploading

\[
\Lambda_m(\tau) = \{\lambda_m(t-\tau), \lambda_m(t+1-\tau), \ldots, \lambda_m(t-1)\}
\]
(20)

where \(\lambda_m(t) \in \{0, 1\}\) indicates whether AUV \(m\) participates in the \(t\) round of gradient upload after controlling the model, \(\lambda_m(t) = 1\) indicates that it participates in this round of the gradient upload, otherwise not.

At one round \(t\), AUV \(m\) checks locally whether Equation (20) is all of 0. If so, AUV \(m\) must be uploaded in this round.

3.4. Latency Model

3.4.1. Local Parameter Calculating Latency

Let \(f_m \in (f_{\text{min}}, f_{\text{max}})\) and \(f_L \in (f_{\text{min}}, f_{\text{max}})\) denote the CPU frequency of the AUV \(m\) and leader AUV \(L\), respectively, where \(f_{\text{min}}\) and \(f_{\text{max}}\) represent the minimum and the maximum CPU frequency, respectively.

The local computation latency on each AUV \(m\) at the \(t\)-th slot can be calculated as

\[
T_{\text{LC}}^m(t) = \frac{N_m \ell_m}{f_m(t)}
\]
(21)
where $c_m$ is the required CPU cycles for training one sample data by backpropagation algorithm [17] at AUV $m$.

### 3.4.2. Uploading Latency

Assume that the local parameter has a counterpart gradient consists of $\alpha$ elements, and each of them has an average quantitative bit number denoted by $\zeta$. The size of the current position information and speed information is $\psi$. Hence, the total data size of each local parameter is $|w_m| = \alpha \zeta$ [17], uploading latency is given by

$$T_{LU}^m(t) = \frac{|w_m| \lambda_m(t)}{R_m^L} + \frac{\psi \lambda_m(t)}{R_m^U}$$

### 3.4.3. Global Parameter Aggregating Latency

At the leader AUV $L$, the global parameter aggregating latency is calculated by

$$T_{GA}^L(t) = \frac{c_0 \sum_{m=1}^{M} |w_m|}{f_L(t)},$$

where $c_0$ is the computational complexity [15] to aggregate the newly updated parameters from follower AUVs, while $|w_m|$ is the data size of the local parameter $w_m$.

### 3.4.4. Global Parameter Updating Latency

The local parameter updating latency is given by

$$T_{GU}^L(t) = \frac{c'_L}{f_L(t)},$$

where $c'_L$ is the computational complexity for performing global parameter updating [17].

### 3.4.5. Downloading Latency

The downloading latency can be calculated as

$$T_{GD}^L(t) = \frac{|w|}{R_m^D} + \frac{\psi}{R_m^D},$$

where $|w|$ is the data size of the global parameter, which is basically similar to the local parameter $|w_m|$.

### 3.4.6. Total Latency

Regardless of extreme cases, there are always devices uploaded in a round, a complete federated learning delay is

$$T(t) = \max_{m \in M} \{ T_{LC}^m + T_{LU}^m \} \max_{m \in M} \{ T_{GA}^L + T_{GU}^L + T_{GD}^L \}.$$
3.5. Energy Consumption Model
3.5.1. Energy Consumption of Follower AUV

The computational energy consumption of the follower AUV is mainly composed of a local parameter calculating and local parameter updating, which can be given by

$$E^{cp}_m(t) = k f_m^c \left( \tau^{lc}_m(t) \right), \quad (28)$$

The communication energy consumption is mainly the local parameter uploading, and it can be calculated by

$$E^c_m(t) = p_m T^u_m(t), \quad (29)$$

3.5.2. Energy Consumption on Leader AUV

The computational energy consumption of the leader AUV is mainly caused by a global parameter aggregating, which is given as

$$E^{cp}_L(t) = k f^c_L \left( \tau^{ga}_L(t) + T^{gl}_L(t) \right), \quad (30)$$

Similarly, the communication energy consumption of leader AUV is mainly due to the global parameter downloading, which can be calculated by

$$E^c_L(t) = p_L T^{gd}_L(t), \quad (31)$$

3.5.3. Total Energy Consumption

$$E(t) = \Phi \left( E^{cp}_m(t) + E^{cp}_L(t) \right) + \chi \left( E^c_m(t) + E^c_L(t) \right) \quad (32)$$

3.6. Problem Formulation

Our goal is to minimize the total cost by optimizing the variables $p_m, f_m, p_L, f_L, \beta$, which are defined as follows.

$$\mathcal{P}1: \min_{p_m, f_m, p_L, f_L, \beta} \text{Cost} = \sum_{i=1}^{N_t} (T(t) + E(t)) \quad (33)$$

$$0 \leq p_m \leq p_{\text{max}}, \quad m \in \mathcal{M}, \quad (34)$$

$$f_{\text{min}} \leq f_m \leq f_{\text{max}}, \quad m \in \mathcal{M}, \quad (35)$$

$$0 \leq p_L \leq p_{\text{max}}, \quad (36)$$

$$f_{\text{min}} \leq f_L \leq f_{\text{max}}, \quad (37)$$

$$0 \leq \beta \leq 1, \quad (38)$$

$$E^{cp}_m + E^c_m \leq E^{thd}_m, \quad m \in \mathcal{M}, \quad (39)$$

$$E^{cp}_L + E^c_L \leq E^{thd}_L, \quad (40)$$

Among them, $N_t$ is the total number of iterations. Equation (33) is the optimization objective, and the total time of each round is minimized by the optimizing power $p_m, p_L$ and CPU frequency $f_m, f_L$, Equations (34) and (35) are the power constraints, and CPU frequency constraints of the AUV $m$, respectively, Equations (36) and (37) are the power constraints and CPU frequency constraints of AUV $L$, respectively, Equation (38) is the lazy node scale factor, and Equations (39) and (40) are the energy constraints of AUV $m$ and AUV $L$, respectively.
4. Algorithm Design

In this section, we consider PPO to deal with \( \mathcal{P} \), since it is non-convex and has high dimensionality.

4.1. Modeling of Deep Reinforcement Learning Environment

An MDP consists of the state space, action space, state transition matrix function, reward function, and discount factor. Thus the optimization problem \( \mathcal{P} \) can be transformed into the following:

State space: For each time slot \( t \), we use \( p_m(t-1), f_m(t-1), p_L(t-1), f_L(t-1), \beta(t-1) \) at the time slot \( t-1 \) to describe the state space.

According to the above, the network state of the agent at time slot \( t \) in this paper can be expressed by

\[
s(t) = (p_m(t-1), f_m(t-1), p_L(t-1), f_L(t-1), \beta(t-1))
\]  

the network state of the agent is given by \( S_t = \{s(t)\} \)

Action space: at each time slot \( t \), \( a(t) \) is composed of the following parts—formally, the action space at time slot \( t \) is denoted by: \( p_m(t), f_m(t), p_L(t), f_L(t), \beta(t) \)

\[
a(t) = (p_m(t), f_m(t), p_L(t), f_L(t), \beta(t))
\]

and the action of the agent is given by \( a_t = \{a(t)\} \).

State transition function: transition probability of the agent at time slot \( t \) can be denoted as \( P_t(s_{t+1} \mid s_t, a_t) \).

Policy: Let \( \pi \) denote the policy function, which is based on the observed state to make decisions and control the action of the agent \( \pi(a \mid s) = P(a \mid s) \).

Reward function: the reward function in this paper is designed for optimizing \( p_m, f_m, p_L, f_L, \beta \); the presented reward function of the agent at one time slot \( t \) is expressed as

\[
r(t) = -(T(t) + E(t))
\]  

Maximizing the cumulative reward obtained by the sequence \( T' \) involves the sum of the rewards obtained at each stage, called \( R_n(T') \). Therefore, the expected reward can be obtained as follows with policy \( \pi \):

\[
R_n(T') = \sum_{\tau=0}^{\infty} \tau T_n(s_{n+\tau}, a_{n+\tau})
\]

where \( \tau \) denotes the discounter factor.

4.2. Proximal Policy Optimization Algorithm

The proximal policy optimization (PPO) is the off-policy model-free reinforcement learning algorithm. OpenAI uses it as the current baseline algorithm, which uses a new class of objective functions and updates parameters in small batches with multiple training steps. To better describe reward and prevent overfitting, we used advantage \( \hat{A}_n \) instead \( R_n(T') \) to evaluate actions

\[
\hat{A}_n = R_n(T') - V_\theta(s_n)
\]  

where \( V_\theta(s_n) \) can be calculated by a value network.

Moreover, our expectation is to update the actor’s policy \( \pi \) to maximize the expected reward. So, we need to use the gradient boosting method to update the network parameters \( \theta \). The gradient solution process is as follows:
G = E [\nabla_\theta \log \pi_\theta(a_t | s_t) \hat{A}_t] 

(46)

PPO2 was inspired by the same question as TRPO: using only the current data to improve the policy as much as possible, without causing a sudden decline in the performance of the policy. Differently from TRPO attempting to solve the problem with a complex second-order approach, PPO2 uses a first-order approach that uses a few other tricks to make the new policy approximate the old one. The essence of the PPO algorithm is to introduce a ratio coefficient to indirectly describe the difference between the new strategy and the old strategy, denoted by \( r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_k}(a_t | s_t)} \); the loss function is defined as follows:

\[
\theta_{k+1} = \arg \max_\theta \left[ \frac{1}{|D_k| T} \sum_{\tau \in D_k} \sum_{t=0}^T \min \left( \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_k}(a_t | s_t)} A_{\pi_k} (s_t, a_t), g(\epsilon, A_{\pi_k} (s_t, a_t)) \right) \right]
\]

(47)

where \( \epsilon \) is a (small) hyperparameter that roughly limits the range of variation of \( r_t(\theta) \). In PPO2, we use the following formula to update the value network strategy:

\[
\phi_{k+1} = \arg \min_\phi \left[ \frac{1}{|D_k| T} \sum_{\tau \in D_k} \sum_{t=0}^T (V_\phi(s_t) - \hat{R}_t)^2 \right]
\]

(48)

The algorithm used in this paper is summarized in Algorithm 1.

**Algorithm 1** Proximal policy optimization clip.

Input: initial policy parameters \( \theta_0 \), initial value function parameters \( \phi_0 \)

for \( k = 1, 2, \ldots \) do

    Run policy \( \pi_\theta \) for \( K \) timesteps, collecting \( s_n, a_n, r_n \)

    Compute return \( \hat{R}_n \) according to Equation (44)

    Compute advantages \( \hat{A}_n \) according to Equation (46)

    Update the policy by maximizing the PPO clip objective according to Equation (47)

    Fit value function according to Equation (48)

end for

5. Simulation Results

This chapter uses the TensorFlow framework in Python to build a federated learning model, using a MNIST (Modified National Institute of Standards and Technology Database) consisting of 42,000 digital images, and retaining 10% of the data as a test set for the global model. At the same time, a three-layer MLP (multilayer perceptron) neural network was created as a model for the classification task. The number of neurons in each layer was 200, 200, and 10, respectively. We used this model to perform a classification task and classify the resulting accuracy and loss function. This section gives a maximum number of convergence rounds of 1000. Assuming \( M = 9 \), the follower AUVs are 20 m apart from each other and form a formation to form a formation and fly at a constant speed in a certain direction, and the leader AUV is 20 m in front of the formation and the center. The rest of the simulation parameters are shown in Table 1.
Table 1. Values of main parameters.

| Parameter | Value   | Parameter | Value   |
|-----------|---------|-----------|---------|
| $k$       | $1.25 \times 10^{-26}$ | $\gamma$ | 0.01    |
| $\delta$  | 3       | $M$       | 20      |
| $B_{UL}$  | 10 kHz  | $f$       | 30 kHz  |
| $B_{UD}$  | 10 kHz  | $s$       | 0.5     |
| $N_m$     | 4224    | $\tau$   | 5       |
| $c_m$     | 10,000  | $f_{\text{max}}$ | 0.4 GHz |
| $c_0$     | 50      | $f_{\text{min}}$ | 0.2 GHz |
| $c_L$     | 50      | $E_{\text{thd}}^\text{m}$ | 0.07 W |
| $|w_m|$    | 1594*64 bit | $E_{\text{thd}}^L$ | 0.8 W |
| $|w|$      | 1594*64 bit | $\Phi$ | 1       |
| $p_{\text{max}}$ | 0.2 W | $\chi$ | 1       |

5.1. The Performance of Each Index in the Gradient Compression Test

In Figure 1, the comparison of communication times among the different schemes versus $\beta$. We can observe that as $\beta$ increases, the follower AUVs that do not participate in communication within the round decrease more, and the corresponding total number of communications increased. From the analysis of Equation (19), the decrease of $\beta$ leads to the increase of the right side of the inequality, resulting in more nodes satisfying Equation (19), the AUV skipping communication rounds increase, and the communication decreases. The experimental results are in good agreement with the theoretical analysis. It can be concluded from Figures 1 and 2 that the federated learning after the control model still converges. However, as it decreases, the number of communications decreases. Although the communication resources are saved, the global model is affected due to the lack of gradients of some follower AUVs in certain rounds, resulting in a decrease in the performance of federated learning. Specifically, the performance is reduced in accuracy, and the experimental results are consistent with the theoretical analysis.

![Figure 1. Comparison of the communication times of different schemes versus $\beta$.](image-url)
In Figure 3, we show the total time comparison of different schemes versus $\beta$. We can observe that as $\beta$ increases, the total time reduces. This is because with $\beta$ increasing, the communications gradually increase, and the model can converge faster with more communications, resulting in a shorter total time.

5.2. The Performance Analysis of the Scheme Proposed in This Paper

To show that the scheme proposed in this paper has the best effect on reducing the training cost, we compare it with other offloading schemes.

**Scheme 1**: the scheme proposed in this paper.
**Scheme 2**: asynchronous federated learning with dynamically optimized $\beta$
**Scheme 3**: asynchronous federated learning with fixed $\beta$.
**Scheme 4**: asynchronous federated learning with LAG.
**Scheme 5**: traditional asynchronous federated learning.

The two sub-graphs in Figure 4 describe the communication times and corresponding accuracy rates of different experimental schemes with different follower AUV numbers. Accuracy will increase. At the same time, a larger communication number corresponds to a higher accuracy rate. When the communication number time is reduced by 710, which is only 21% of the traditional asynchronous federated learning, the accuracy rate is only reduced by 0.03%. This shows that we greatly reduced the communication number, but the accuracy basically did not decrease, indicating that our proposed scheme has good results.

![Figure 4](image.png)

**Figure 4.** The relationship between communication times and accuracy. (a) Comparison of the communication times of different schemes versus the number of follower AUVs. (b) Comparison of the accuracy of different schemes versus the number of follower AUVs.

In Figure 5, we show the accuracies of different control models versus the number of follower AUVs. Compared with the work of [19], the control model proposed in this paper increases the accuracy. This shows that the improved control model in this paper is effective.

![Figure 5](image.png)

**Figure 5.** Comparison of the accuracies of different control models versus the number of follower AUVs.
In Figure 6, we show the cost comparison of different schemes versus the number of follower AUVs. With the increase in the follower AUV, the cost gradually increases. As can be seen from the figure, the scheme proposed in this paper has a smaller cost and can save more resources.

![Figure 6. Comparison of the costs of different schemes versus $\beta$.](image)

5.3. The Performance Analysis of the PPO2 Algorithm

In Figure 7, we compare the performance of state-of-the-art algorithms in coping with problem $P_1$, including GA, PSO, AC, DDPG, and our employed PPO2 algorithm. We can observe that PPO2 has a smaller cost under the same number of iterations, which shows that this algorithm is better than other algorithms.

![Figure 7. Comparison of the profits of different algorithms.](image)

6. Conclusions

In order to reduce the constraints of underwater scarce communication resources on AUV swarm machine learning, we designed an asynchronous federated learning method. By constructing the optimization problem of minimizing the weighted sum of delay and
energy consumption, the AUV CPU frequency and signal transmission power were jointly optimized. In order to solve this complex optimization problem of high-dimensional non-convex time series accumulation, we transformed the problem into an MDP and used the PPO2 algorithm to solve this problem. Finally, we carried out some experiments to verify the effectiveness of the proposed scheme.

**Author Contributions**: This manuscript was designed and written by Z.M., who conceived the main idea of this study. Z.M. wrote the program and completed all experiments. Z.L. supervised the research and contributed to the proposal and improvement of the algorithms. X.H., J.D., J.C. and W.W. contributed to the work. All authors have read and agreed to the published version of the manuscript.

**Funding**: This work was supported in part by the National Natural Science Foundation China under project 61971257, in part by the Young Elite Scientist Sponsorship Program by CAST under Grant 2020QNRC001, and in part by the National Natural Science Foundation China under project 61673310. (Corresponding author: Zhi Li).

**Institutional Review Board Statement**: Not applicable.

**Informed Consent Statement**: Not applicable.

**Data Availability Statement**: Not applicable.

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

**Abbreviations**

The following abbreviations are used in this manuscript:

FL federated learning  
AUVs autonomous underwater vehicles  
UIoT underwater Internet of Things  
PPO2 proximal policy optimization 2  
FEDSGD federated stochastic gradient descent  
DNN deep neural network  
ANN artificial neural network

**References**

1. Fang, Z.; Wang, J.; Jiang, C.; Wang, X.; Ren, Y. Average Peak Age of Information in Underwater Information Collection With Sleep-Scheduling. *IEEE Trans. Vehicular Technol.* **2022**, *71*, 10132–10136. [CrossRef]  
2. Guan, S.; Wang, J.; Jiang, C.; Duan, R.; Ren, Y.; Quek, T.Q.S. MagicNet: The Maritime Giant Cellular Network. *IEEE Commun. Mag.* **2021**, *59*, 117–123. [CrossRef]  
3. Fang, Z.; Wang, J.; Jiang, C.; Zhang, Q.; Ren, Y. AoI-Inspired Collaborative Information Collection for AUV-Assisted Internet of Underwater Things. *IEEE Internet Things J.* **2021**, *8*, 14559–14571. [CrossRef]  
4. Fang, Z.; Wang, J.; Du, J.; Hou, X.; Ren, Y.; Han, Z. Stochastic Optimization-Aided Energy-Efficient Information Collection in Internet of Underwater Things Networks. *IEEE Internet Things J.* **2022**, *9*, 1775–1789. [CrossRef]  
5. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics—AISTATS*, Ft. Lauderdale, FL, USA, 20–22 April 2017.  
6. Poudel, S.; Moh, S. Medium Access Control Protocols for Unmanned Aerial Vehicle-Aided Wireless Sensor Networks: A Survey. *IEEE Access* **2019**, *7*, 65728–65744. [CrossRef]  
7. Shen, C.; Shi, Y.; Buckham, B. Integrated Path Planning and Tracking Control of an AUV: A Unified Receding Horizon Optimization Approach. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 1163–1173. [CrossRef]  
8. Yan, J.; Yang, X.; Luo, X.; Chen, C. Energy-Efficient Data Collection Over AUV-Assisted Underwater Acoustic Sensor Network. *IEEE Syst. J.* **2018**, *12*, 3519–3530. [CrossRef]  
9. Cai, L.; Zhou, G.; Zhang, S. Multi-AUV Collaborative Hunting Method for the Non-cooperative Target in Underwater Environment. In *Proceedings of the 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM)*, Singapore, 18–20 July 2018; pp. 1–5. [CrossRef]  
10. Cui, R.; Li, Y.; Yan, W. Mutual Information-Based Multi-AUV Path Planning for Scalar Field Sampling Using Multidimensional RRT*. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 992–1004. [CrossRef]  
11. Noguchi, Y.; Maki, T. Path Planning Method Based on Artificial Potential Field and Reinforcement Learning for Intervention the AUVs. In *Proceedings of the 2019 IEEE Underwater Technology (UT)*, Kaohsiung, Taiwan, 16–19 April 2019; pp. 1–6. [CrossRef]
12. Huang, H.; Zhu, D.; Yuan, F. Dynamic task assignment and path planning for multi-AUV system in 2D variable ocean current environment. In Proceedings of the 24th Chinese Control and Decision Conference (CCDC), Taiyuan, China, 23–25 May 2012; pp. 3660–3664. [CrossRef]

13. Cao, X.; Sun, C. Multi-AUV cooperative target hunting based on improved potential field in underwater environment. In Proceedings of the 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), Nanjing, China, 18–20 May 2018; pp. 118–122.

14. Van, N.T.T.; Luong, N.C.; Nguyen, H.T.; Shaohan, F.; Niyato, D.; Kim, D.I. Latency Minimization in Covert Communication-Enabled Federated Learning Network. *IEEE Trans. Vehicular Technol.* **2021**, *70*, 13447–13452. [CrossRef]

15. Lu, Y.; Huang, X.; Zhang, K.; Maharjan, S.; Zhang, Y. Communication-Efficient Federated Learning and Permissioned Blockchain for Digital Twin Edge Networks. *IEEE Internet Things J.* **2021**, *8*, 2276–2288. [CrossRef]

16. Chen, M.; Poor, H.V.; Saad, W.; Cui, S. Convergence Time Optimization for Federated Learning Over Wireless Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 2457–2471. [CrossRef]

17. Ren, J.; Yu, G.; Ding, G. Accelerating DNN Training in Wireless Federated Edge Learning Systems. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 219–232. [CrossRef]

18. Zhang, J.; Simeone, O. LAGC: Lazily Aggregated Gradient Coding for Straggler-Tolerant and Communication-Efficient Distributed Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 962–974. [CrossRef] [PubMed]

19. Chen, T.; Sun, Y.; Yin, W. Communication-Adaptive Stochastic Gradient Methods for Distributed Learning. *IEEE Trans. Signal Process.* **2021**, *69*, 4637–4651. [CrossRef]

20. Sun, J.; Chen, T.; Giannakis, G.B.; Yang, Q.; Yang, Z. Lazily Aggregated Quantized Gradient Innovation for Communication-Efficient Federated Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 2031–2044. [CrossRef] [PubMed]

21. Jensen, F.B.; Kuperman, W.A.; Porter, M.B.; Schmidt, H.; Tolstoy, A. *Computational Ocean Acoustics*; Springer: Berlin/Heidelberg, Germany, 2011; Volume 2011.

22. Stojanovic, M. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mob. Comput. Commun. Rev.* **2007**, *11*, 34–43. [CrossRef]