Machine Learning based Semantic Communication Systems for 6G Three-dimensional Communication Networks

Guhan Zheng

School of Computing and Communications

Lancaster University

A thesis submitted for the degree of

Doctor of Philosophy

January, 2024

This thesis is dedicated to my loving parents.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the references.

Guhan Zheng

Abstract

The sixth generation (6G) wireless communication is anticipated as a three-dimensional (3D) network with full support of aerial edge and space edge. Moreover, semantic communication (SemCom) based on machine learning (ML) is also considered a significant enabling technology for 6G systems. Nevertheless, integrating SemCom into future 3D networks introduces emerging semantic coder updating requirements and new functional challenges considering, e.g. latency, energy, and privacy. Motivated by the above observations, in this thesis, the challenges of SemCom in various 6G edge-enable network architectures are investigated.

Firstly, a terrestrial vehicular SemCom system is investigated for vehicle task offloading in vehicular networks (VNs). A novel mobility-aware split-federated with transfer learning (MSFTL) framework for SemCom coder updating is then proposed. Moreover, to incorporate vehicle mobility and training delays I propose a high-mobility training resource optimisation mechanism based on a Stackelberg game for MSFTL.

Secondly, an air-terrestrial SemCom system is proposed for energy-efficient implementation of SemCom in aerial-aided edge networks (AENs). An energy-efficient game theoretic incentive mechanism (EGTIM) is proposed for improving the energy efficiency of the AEN for SemCom. To update SemCom coders accurately and efficiently in AENs, I further present a game theoretic efficient distributed learning (GEDL) framework based on the renewed EGTIM.

Finally, a space-air-terrestrial (SAT) SemCom system is proposed for the computation offloading of resource-limited users in SAT networks. An adaptive pruning-split federated learning (PSFed) method for updating the SemCom coder is then proposed. Furthermore, the users processing computational tasks strategy in presented systems is formulated as an incomplete information mixed integer nonlinear programming (MINLP). A new computational task processing scheduling (CTPS) mechanism is also proposed based on the Rubinstein bargaining game.

Publications

G. Zheng, Q. Ni, K. Navaie and H. Pervaiz, ”Semantic Communication in Satellite-borne Edge Cloud Network for Computation Offloading,” *IEEE Journal on Selected Areas in Communications*, Accepted to appear.

G. Zheng, Q. Ni, K. Navaie, H. Pervaiz and C. Zarakovitis, ”A Distributed Learning Architecture for Semantic Communication in Autonomous Driving Networks for Task Offloading,” *IEEE Communications Magazine*, vol. 61, no. 11, pp. 64-68, November 2023.

G. Zheng, Q. Ni, K. Navaie, H. Pervaiz, G. Min, A. Kaushik, and C. Zarakovitis, ”Mobility-aware Split-Federated with Transfer Learning for Vehicular Semantic Communication Networks,” *IEEE Internet of Things Journal*, Accepted to appear.

G. Zheng, Q. Ni, K. Navaie, H. Pervaiz, A. Kaushik and C. Zarakovitis, ”Energy-efficient Semantic Communication for Aerial-aided Edge Networks,” submitted to *IEEE Transactions on Green Communications and Networking.*

G. Zheng, Q. Ni, K. Navaie, H. Pervaiz and C. Zarakovitis, ”Efficient Pruning-Split LSTM Machine Learning Algorithm for Terrestrial-Satellite Edge Network,” *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, Seoul, Korea, Republic of, 2022, pp. 307-311.

Acknowledgements

I would like to acknowledge my supervisors. From the selection of the research topic to the final draft, Prof. Qiang Ni and Prof. Keivan Navaie greatly respected my ideas and opinions. Simultaneously, they provided me with meticulous care, continuous support and patient guidance. Whenever I faced challenges in my research, they always listened patiently and offered me advice and assistance. This thesis would not have been achievable without my supervisors. I can't imagine better supervisors.

I must also convey my appreciation to Dr. Haris Pervaiz for his professional advice and invaluable support for me during his time at Lancaster.

Furthermore, I wish to express my gratitude to all academic and administrative staff at the School of Computing and Communications, and all my friends in Lancaster. Especially Dr. Zhengxin Yu and Dr. Haitao Zhu. They have provided me with immense support and assistance, particularly during the challenging period of the COVID-19 pandemic, for my overall well-being and successful completion of my Ph.D.

Last but not least, my heartfelt appreciation also goes to my parents for their love and unwavering support.

Contents

Declaration **iii**

Abstract **iv**

Publications **v**

Acknowledgements **vi**

List of Tables **x**

List of Figures **xii**

List of Acronyms **xv**

List of Mathematical Operators and Main Symbols **xvii**

1 Introduction **1**

1.1 Motivations 1

1.2 Thesis contents and contribution 3

1.3 Thesis outline 7

2 Theoretical Background and Literature Review **8**

2.1 Semantic communication 8

2.2 Edge cloud in 3D networks 10

2.3 Distributed learning 12

2.4 SemCom in networks 13

2.5 Summaries 15

3 Terrestrial Vehicular SemCom System 16

3.1 Introduction 16

3.2 System model 17

3.2.1 Vehicle SemCom model 18

3.2.2 Computing and communication model 18

3.3 MSFTL for vehicle SemCom 20

3.3.1 Transfer learning for vehicle SemCom network 20

3.3.2 MSFTL design 21

3.3.3 Comparison of computing and communication overhead 24

3.4 Stackelberg game based resource optimisation mechanism 26

3.4.1 Game design at the vehicles 27

3.4.2 Game design at the TEC 29

3.4.3 Optimal solutions and equilibrium analysis 30

3.4.4 MSFTL 31

3.5 Simulation results 31

3.5.1 Optimisation mechanism 37

3.6 Summaries 39

4 Air-terrestrial SemCom System 41

4.1 Introduction 41

4.2 System model 42

4.3 Stackelberg game theoretic mechanism design 45

4.3.1 Game at the AEC 45

4.3.2 Game at TECs 47

4.3.3 Nash equilibrium for the game 49

4.4 Efficient distributed Learning Design 51

4.5 Simulation results 54

4.5.1 EGTIM 55

4.5.2 GEDL 57
4.6 Summaries 61

5 Space-air-terrestrial SemCom System 63
5.1 Introduction 63
5.2 System model 64
5.2.1 System description 64
5.2.2 Computing models 66
5.2.3 Communication models 67
5.2.4 Path loss model 69
5.2.5 Semantic coder training model 71
5.3 Semantic coders: updating 71
5.4 Semantic coders: in service 75
5.4.1 Problem for computational task processing 75
5.4.2 CTPS 79
5.4.3 First step of the CTPS mechanism 79
5.4.4 Second step of the CTPS mechanism 80
5.5 Simulation results 84
5.5.1 Simulation setting 84
5.5.2 Performance evaluation of the proposed PSFed 86
5.5.3 Performance evaluation of the proposed CTPS 90
5.6 Summaries 95

6 Conclusions and Future Work 96
6.1 Summary 96
6.2 Future work 97

References 99

List of Tables

3.1 The setting of the CAE in the proposed semantic network framework. 32

5.1 The setting of the CAE 85

5.2 Rainfall coefficients 86

5.3 Simulation parameters 86

List of Figures

1.1 Semantic versus conventional communication transmission systems. . 2

2.1 Autoencoder model. 9

2.2 SemCom coder updating in edge networks based on central learning. . 11

2.3 SemCom coder updating in edge networks based on federated learning. 13

3.1 Vehicles in the network. 19

3.2 The functional block diagram of the proposed MSFTL. 23

3.3 Convergence speed comparison of different frameworks. 33

3.4 Accuracy of different frameworks. 34

3.5 Computing cost of different frameworks. 35

3.6 Communication cost of different frameworks. 36

3.7 Accuracy of different frameworks with sparse samples. 37

3.8 Reward impact on training data number. 38

3.9 Total training time versus various residence time. 39

4.1 Proposed system model. 43

4.2 The process of proposed GEDL. 52

4.3 NE existence under the proposed EGTIM. 55

4.4 Energy saving of proposed EGTIM in various scenarios. 56

4.5 Effect of different CPU-cycle on providing task size. 57

4.6 The accuracy of various training frameworks with the AEC input samples grows. 58

4.7 Convergence speed of different training frameworks. 59

4.8 Energy saving of proposed GEDL in various scenarios. 60

4.9 The impact of β value on energy saving. 61

5.1 The proposed SemCom-SEC framework. 65

5.2 Framework of the proposed PSFed in one communication round. . . . 72

5.3 Proposed CTPS mechanism. 81

5.4 Convergence speed of various learning algorithms with different datasets. 88

5.5 Communication cost of various learning approaches. 89

5.6 Accuracy of various learning algorithms with different datasets. . . . 90

5.7 Privacy leakage of various learning approaches. 92

5.8 The processing cost of the varying number of users. 92

5.9 The processing cost of a single user. 93

5.10 The usefulness of SemCom in the network. 94

5.11 Impact of α and β on strategy developing. 94

List of Acronyms

3D Three-dimensional.

6G 6^{th} generation.

AEC Aerial edge cloud.

AEN Aerial-aided edge networks.

AWGN Additive white Gaussian noise.

BS Base station.

CAE Convolutional autoencoder.

CNN Convolutional neural network.

CTPS Computational task processing scheduling.

DL Deep learning.

DLJSC DL-based joint source-channel.

DRL Deep reinforcement learning.

EGTIM Energy-efficient game theoretic incentive mechanism.

FL Federated learning.

GEDL Game theoretic efficient distributed learning.

IID Independent and identically distributed.

J Joule.

KKT Karush-Kuhn-Tucker.

LEO Low earth Orbit.

MEC Multi-access edge cloud.

MINLP Incomplete information mixed integer nonlinear programming.

ML Machine learning.

MSE Mean squared error.

MSFTL Mobility-aware split-federated learning with transfer learning.

NE Stackelberg equilibrium.

OFDMA Orthogonal frequency division multiple access.

PSFed Pruning-split federated learning.

PSNR Peak Signal-to-Noise Ratio.

QoS Quality of service.

RSU Roadside unit.

SAT Satellite-air-terrestrial.

SEC Satellite edge cloud.

SemCom Semantic communication.

SemCom-SEC SemCom-assisted SEC.

SL Split learning.

TEC Terrestrial edge cloud.

TL Transfer learning.

TST Terrestrial-station-terminal.

UAV Unmanned aerial vehicles.

VN Vehicular network.

WNO Wireless network operator.

List of Mathematical Operators and Symbols

Mathematical Operators

$\forall$	Universal quantifier (For all)
$\frac{\partial f(x,y)}{\partial x}$	Second-order partial derivative of function f with respect to x
$\frac{\partial^2 f(x,y)}{\partial x^2}$	Partial derivative of function f with respect to x
$\int$	Integration of sets
$\ln$	Natural logarithm
$\log_i$	Logarithm base i
erf	Gauss error function
$\max$	Maximum function
$\min$	Minimum function
$\sum$	Summation operation
$A(\cdot)$	Autoencoder calculation process
$s.t$	Subject to

Symbols

$x^{n,m}$	ML model input sample
$\tilde{x}^{n,m}$	ML model output sample
f	CPU-cycle frequency
B	Bandwidth (bps)
p	Power (Watt)
ε	Electronics related energy factor
U	Utility function
C	Computing/communication cost
σ_0	Additive white Gaussian noise power (Watt)
h	Satellite altitude (meters)
PL	Path loss
L	Training loss

Chapter 1

Introduction

1.1 Motivations

The 6^{th} generation (6G) wireless communication is considered a three-dimensional (3D) communication network fully assisted by edge cloud facilities [1], [2]. The aerial facilities and satellites with edge clouds, i.e., aerial edge clouds (AECs) and satellites edge clouds (SECs), are anticipated to provide abundant storage and computing resources to subscribers alongside the terrestrial edge clouds (TECs). Subscribers are allowed to access these edge facilities to offload computationally sensitive tasks for rapid processing or acquire massive image/video information etc. [3]. However, since wireless physical layer capacity is approaching the Shannon limit, current wireless technologies are becoming increasingly insufficient to satisfy such a sophisticated, data traffic and diverse offloading need in future 6G 3D networks [4]. How to improve the communication efficiency and Quality of service (QoS) for future communication systems thus become an emerging challenge in the development of 6G-enabled networks.

Semantic communication (SemCom) is a new intelligent communication paradigm and is considered a promising solution for 6G to address this challenge [5]. Different from the conventional Shannon paradigm [6], SemCom is a genuinely intelligent system that only selects the necessary information to be transmitted. It concentrates on the meaning of the information transmitted and ignores irrelevant information by employing deep learning (DL) approaches [7]. Using this approach, the network spectral efficiency is significantly reduced, thereby improving the performance of the communication network.

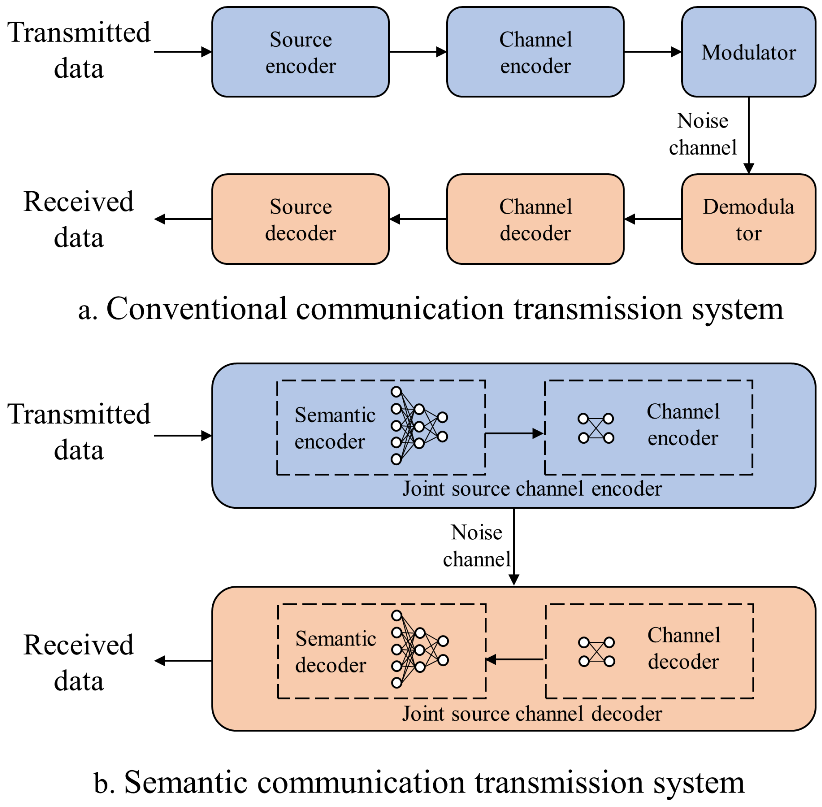


Figure 1.1: Semantic versus conventional communication transmission systems.

Generally, the coder in SemCom is designed as a DL-based joint source-channel (DLJSC) coder to substitute the conventional transmission coder [8] (Figure 1.1). However, in this approach, the DLJSC encoder and decoder are deployed in the transmitter and receiver separately but are required to be trained for particular transmission contents together. This introduces the question of SemCom deployment and utilisation in practical networks. Moreover, in practice, the DLJSC coder model needs to continue learning and updating on previously untrained content to ensure providing a consistent QoS [9]. This presents several different emerging challenges for different future 3D networks, e.g. collaboration coder updating of transmitter and receiver, the dynamism of some networks, and different users' different encoder models. Nevertheless, the existing SemCom systems and distributed learning frameworks for semantic communications (see, e.g., [10]–[12]) in generic networks are however not automatically applicable to the various complex 3D networks. Designing efficient SemCom systems and distributed learning methods for updating the semantic coders in 6G networks thus is essential.

In addition to the above, SemCom alters the transmission paradigm of conventional networks by increasing the computational load while reducing the communication load. It changes the existing pattern of communication and computation resource utilisation. This new communication paradigm means a new trade-off to be made in future SemCom-assisted networks in terms of technical factors such as delays, energy cost and privacy.

Therefore, in this thesis, the objectives are to design efficient SemCom systems to address different unique challenges for various 3D networks, i.e., terrestrial vehicular networks, air-terrestrial networks and space-air-terrestrial networks. In addition, novel SemCom coder updating learning frameworks are investigated while new and technical challenges of SemCom in 6G networks are also considered, e.g. delay, energy and privacy.

1.2 Thesis contents and contribution

Motivated by the discussions aforementioned, this thesis focuses on the unique challenges of SemCom in various future wireless networks. Specifically, by designing SemCom systems and novel ML frameworks, the SemCom coders enable updating and employment efficiently. Furthermore, economic game theories are utilised to analyse and investigate the SemCom functional challenges in various networks in terms of delay, energy and privacy. The main contributions of this thesis are summarised as the following.

Chapter 2 provides the background theory and literature related to the system design of this thesis. The background knowledge of SemCom, edge cloud and 3D networks is introduced first. In addition, the basic theory of the technological areas utilised in this thesis is also presented, i.e., collaborative learning and game theories. The existing studies of SemCom in networks are then introduced to help readers understand the research background.

In Chapter 3, the terrestrial vehicular SemCom system is investigated. A mobility-aware split-federated learning framework is then proposed for SemCom coder updating to solve the unique challenges of SemCom in vehicular networks. Moreover, although an un-updated model causes degradation in accuracy for new transmission tasks, the un-updated encoder model can be exploited to increase training efficiency and decrease the computing and communications cost of distributed training. A novel transfer learning (TL) [13] paradigm for vehicular SemCom is further proposed to be integrated into the presented framework by employing part of the un-updated encoder. The proposed mobility-aware split-federated with transfer learning framework is referred to as MSFTL. In addition, a high-mobility training energy optimisation mechanism for MSFTL is presented based on the Stackelberg game. The main contributions of this chapter are as follows:

- The terrestrial vehicular SemCom system is investigated and a novel MSFTL framework for vehicular semantic communication networks is proposed. The proposed model splits the coder into four separate components for training. The vehicle only needs to train parts of the coder to reduce the cost of computing. MSFTL addresses unique challenges for semantic communications in vehicular networks that were not addressed by the existing learning framework for semantic communication networks.

- A new TL-based learning approach is presented in the developed MSFTL. Here, by utilising the part of the un-updated semantic encoder model, the MSFTL increases the convergence speed and accuracy. It decreases the training computing and communication cost. This approach also reduces storage load and performs well on a few sample learning scenarios.

- A Stackelberg game-based energy optimisation mechanism is developed to further reduce the training energy cost and optimise the proposed framework. The most appropriate amount of training data is selected for each vehicle and the entire network. It jointly considers factors such as vehicle residence time, computational load, and communication overhead.

In Chapter 4, a novel air-terrestrial SemCom system is proposed for aerial-aided edge networks (AENs). The resource allocation problem during SemCom usage is then discussed. A new energy-efficient game theoretic incentive mechanism (EGTIM) based on the proposed system is presented to optimise the network energy efficiency in a fair way. In addition, a game theoretic efficient distributed learning (GEDL) framework is designed for semantic coders updating in AENs. It updates the proposed EGTIM and integrates EGTIM with traditional distributed learning methods to accurately and efficiently update the semantic coder with respect to energy consumption. The major contributions of this chapter are summarised as follows:

- A novel air-terrestrial SemCom system to support AENs is proposed. In this system, AECs and TECs provide edge services to users via employed ML-based semantic coders. Moreover, it enables edge devices to schedule the processing locations of computational tasks due to semantic communication intelligently to improve the energy efficiency of the AEN. The AENs' spectral efficiency and the QoS thus can be improved.

- In particular, a new EGTIM in the proposed SemCom system is presented to further improve the energy efficiency of AENs. The computational and communication workload of the AEC and TECs to perform semantic communication are developed as a Stackelberg game. It is designed to maximise the energy efficiency of the AEN while proportional fairness maximising the service revenue of each edge device in the network.

- A GEDL framework is proposed for semantic coder updating in AENs. It is based on our designed renewed EGTIM for semantic coder updating. Compared to federated learning (FL), it significantly improves the semantic coder accuracy in Non-IID scenarios and improves the training energy efficiency by retraining the model after federated aggregation in the AEC.

In Chapter 5, the SemCom system for space-air-terrestrial (SAT) networks is designed. A new SemCom-assisted SEC (SemCom-SEC) framework is put forward for computation offloading by terrestrial users. The proposed approach divides the SemCom service into two scenarios: in-maintenance (where semantic coders need updating) and in-service (where trained semantic coders are used for offloading computations). In the in-maintenance scenario, the real-time update of deployed semantic coders in SemCom-SEC is explored. Following this, a pruning-split federated learning (PSFed) method is introduced to update semantic coders while taking into account offloading quality of service (QoS) and ensuring privacy. In the in-service scenario, the challenge of computational task processing for terrestrial users under the new SemCom paradigm is examined. A novel computational task processing scheduling (CTPS) mechanism is then suggested, based on the Rubinstein bargaining game, which aims to minimize users' processing delay and energy consumption while safeguarding their privacy. The main contributions of this chapter are summarised as follows:

- The SemCom and SEC networks are integrated and a novel SemCom-SEC framework enabling task offloading for under-served users is proposed. In the proposed framework, the SemCom coders are deployed on both the TSTs and satellites. The SemCom-SEC takes into account various user task-processing approaches and access modalities. The user's computational tasks can be either performed locally, at SEC or in the core cloud server. Moreover, users have the option to access the LEO satellites either directly or via the semantic encoder-equipped TST.

- A PSFed approach for SemCom coder updating for the SemCom-SEC framework enabling computation offloading is then presented. PSFed adaptively "splits" and "prunes" the semantic coders for federated aggregation subject to various users' personalised conditions. In contrast to the conventional "split" and "prunes" models, the semantic coder model components remain intact after updating. PSFed reduces the consumption of training communication resources and improves the privacy of the trained encoder while enhancing the training convergence speed and model accuracy.
- A novel CTPS mechanism is proposed by jointly considering user privacy, delay, energy consumption and fairness to solve the new incomplete information task processing scheduling problem in SemCom-SEC. The CTPS performs in two steps. A game theoretic model is first designed to convert this mixed integer nonlinear programming (MINLP) problem from an incomplete information problem due to privacy concerns to a complete information problem. In the second step, the converted complete information MINLP problem is decomposed and solved by adopting the Lagrangian dual decomposition method etc.

1.3 Thesis outline

The rest of the thesis is organised as the following. Chapter 2 presents the background knowledge of this thesis with a brief literature review. In Chapter 3, Chapter 4, and Chapter 5, SemCom systems for terrestrial vehicular networks, air-terrestrial networks and SAT networks are presented separately. Finally, conclusions and future works are discussed in Chapter 6.

Chapter 2

Theoretical Background and Literature Review

2.1 Semantic communication

The recent development of ML technologies enabled the integration of semantic communication into 6G as a promising solution for improving channel spectrum efficiency. In contrast to the Shannon paradigm that focuses on the accuracy of symbol transmission, semantic communication exploits ML to extract the actual meaning of information to reduce the transmission information quantity [7]. In semantic communication, the conventional coder is substituted by a semantic DLJSC that compresses and transmits semantic information, where the coder is an ML-based Autoencoder model [8].

The Autoencoder model (Figure 2.1) is a type of ML used for unlabeled data, i.e., unsupervised learning. It learns the implicit features, i.e., semantic features, of the input data, which is called coding from the encoder, and reconstructs the original input data with the learned new features, which is called decoding from the decoder. The Autoencoder thus can function as a feature/semantic extractor.

It can be seen in Figure 2.1, there are three main components required for the construction of an Autoencoder, i.e., encoder, decoder and loss function. Encoder

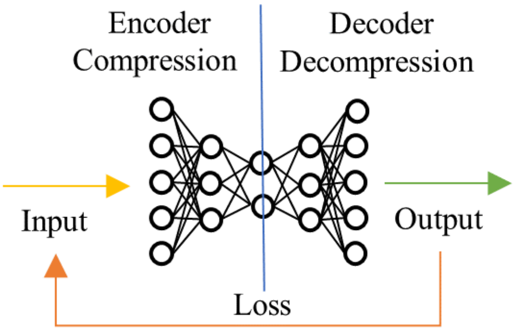


Figure 2.1: Autoencoder model.

and decoder are parametric equations that form the Autoencoder model. Normally, they are based on neural networks, which are derivable with respect to the loss function based on stochastic gradient descent etc. Furthermore, the loss function is a metric that measures the volume of information lost after compression and decompression. Mathematically, the input convert process of Autoencoder can be expressed as:

$$\tilde{x_1} = A_1(x), \tag{2.1}$$

$$\tilde{x_2} = A_2(\tilde{x_1}), \tag{2.2}$$

where $A_1(\cdot)$ is the coding and $A_2(\cdot)$ is decoding, x is the input of the autoencoder and $\tilde{x_1}$ is the compression output of the encoder, i.e., feature/semantic information. Moreover, $\tilde{x_2}$ is the output recovered by decompression through the decoder. The loss function thus is the comparison between x and $\tilde{x_2}$.

In SemCom studies, the encoder part of the Autoencoder can be deployed at the transmitter and the decoder part can be deployed at the receiver. Encoders and decoders can also adopt different neural network models. The transmitter merely transmits the semantic feature of the input encoded by the ML-based encoder to the ML-based receiver decoder for recovery. The number of transmission bits is significantly reduced. SemCom thus goes beyond the Shannon capacity limit by shifting the proportion of the work to computational resources from communication and significantly increases the spectral efficiency [14].

To do this, various semantic communication studies have been developed for image transmission [15]–[18], text transmission [10], [19], [20], video transmission [21],[22], speech [23], and visual question answering transmission [24]. These efforts demonstrated the excellent performance of the SemCom systems in upgrading communication efficiency and transmission accuracy. The SemCom is hence considered one of the emerging and promising techniques for 6G.

2.2 Edge cloud in 3D networks

Next-generation communication networks are considered to be not only networks supported by terrestrial cellular devices but also 3D networks coordinated by space (satellites), air (unmanned aerial vehicles (UAVs), airships, and balloons) and terrestrial communication devices [25]. The development of 3D communications is necessary for the following reasons:

1. The service area of terrestrial cellular networks generally cannot reach 100% global coverage. For instance, in mountainous areas and deserts, infrastructures are difficult to deploy.

2. Natural disasters may destroy the communication entities, resulting in complete destruction of the terrestrial facilities. In this case, it is crucial to use space and air networks to improve the robustness of the entire communication system and to react quickly to the information.

3. Terrestrial facilities' service capabilities are subject to the constraints of limited local resources such as spectrum, power or cache capacity, thus requiring flexible equipment assistance.

Therefore, the integration of space, air and terrestrial networks is necessary. It extends the coverage of the service area, provides QoS-guaranteed services, balances inefficient communication resource allocation, and delivers content to the edge of the network.

Deploying cloud facilities at these edge devices of the 6G networks, i.e., edge cloud, is also emerging as one of the key techniques for next-generation wireless communication systems [26]. Cloud facilities with powerful computation and storage capabilities are devolved to the edge of the network, allowing for providing subscribers with abundant cloud computational resources. Subscribers are allowed to access these edge facilities to offload computationally sensitive tasks for rapid processing or acquire massive image/video information etc. [3].

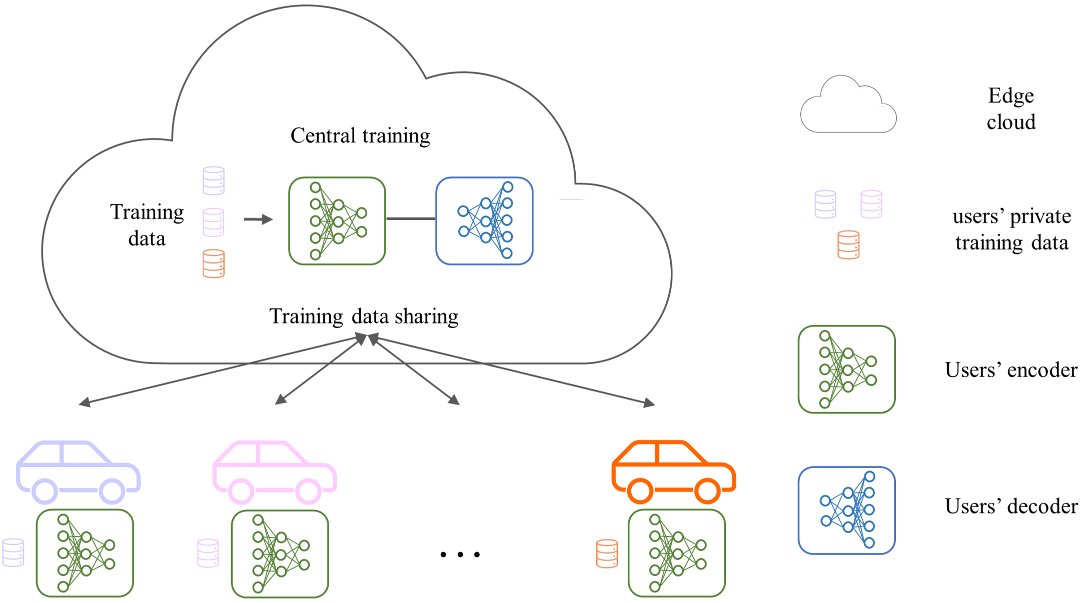


Figure 2.2: SemCom coder updating in edge networks based on central learning.

In terrestrial networks, the TECs can be deployed on the base stations (BSs) and roadside units (RSUs) etc. Vehicles can access these resources by offloading their tasks (e.g. object/image recognition and processing tasks, etc.) to the TEC in real-time via a communication link. The aerial facilities, e.g. UAVs, airships, and balloons, with edge clouds, i.e., AECs, are anticipated to provide abundant storage and computing resources to subscribers alongside the TECs.

In addition, subscribers located in remote areas or disaster zones might not be able to connect to TEC infrastructures. The arrival cost of AEC facilities is also prohibitive. Alternatively, such under-served users may offload their computationally intensive tasks to remote core cloud servers via Geosynchronous

Equatorial Orbit (GEO) or Medium Earth Orbit (MEO) satellites. In addition to the costs, the corresponding propagation latency to and from the satellite platforms however impedes the delay requirements of these users. Using Low Earth Orbit (LEO) satellites can partly address this issue by providing lower propagation latency as their orbits are much closer to the ground compared to GEO and MEO satellites. Comparing to GEO and MEO, constellations of LEO satellites also provide low-cost, high-throughput services and extensive radio coverage. To further reduce the propagation delay, the SEC setting was proposed, where the offloaded processing is conducted on board the LEO satellite, hence reducing the propagation delay by a factor of 2 [27], [28].

2.3 Distributed learning

Central learning (Figure 2.2) is a conventional collaborative learning approach developed based on the conventional approach of training neural networks on a single server. The training data from distributed users are collected by a central server, e.g. edge cloud. Subsequently, all training data on the central server is integrated and used as input to jointly train an ML model. The trained model is then returned to the participating users. Since in central learning, the training data are trained directly by the ML model, it is therefore capable of obtaining higher accuracy relative to other distributed learning methods.

However, central learning is not applicable to edge computing [29]. Because moving heavy training data over the network implies significant transmission delays, let alone potential privacy breaches during training data transmission. Nevertheless, allowing distributed users to update/train the ML model locally would suffer from insufficient performance, energy and few-shot samples.

FL (Figure 2.3) [30] is a promising distributed learning framework for collaborative training in edge cloud networks. In each training epoch, distributed users first train the entire model on the user side using their individual training data and then

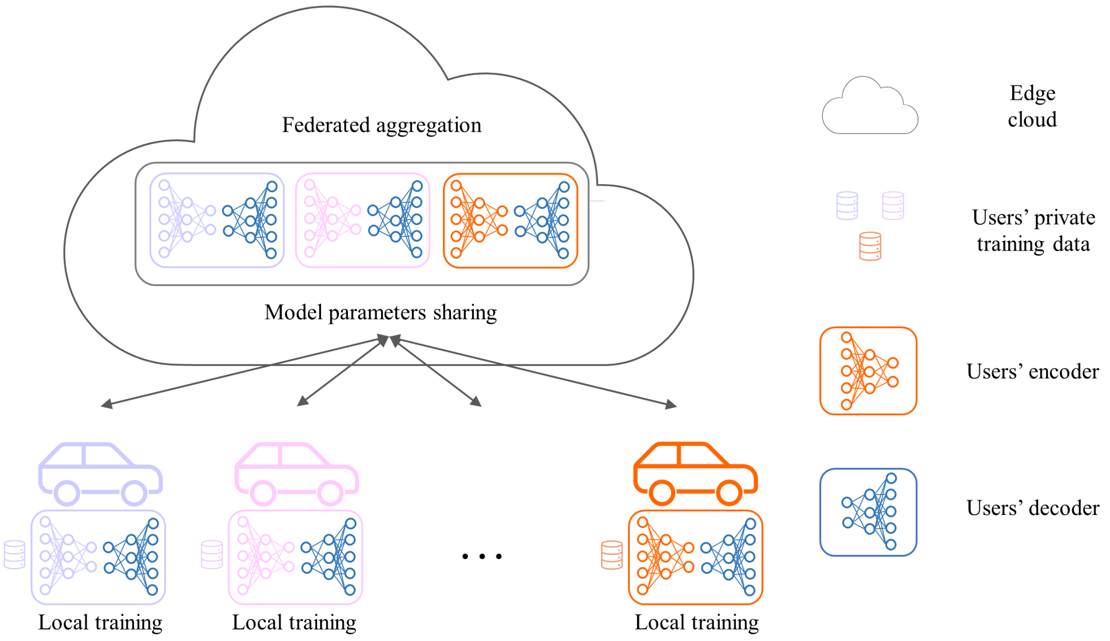


Figure 2.3: SemCom coder updating in edge networks based on federated learning.

upload the model weights to a central server for aggregation. The aggregated model is then sent back to the participating users. This enables individual clients to keep their private training data locally, hence preserving their data privacy and avoiding the problems associated with centralised data collection.

2.4 SemCom in networks

Different from point-to-point SemCom techniques research, the existing system designing and collaborative learning frameworks for SemCom in terrestrial networks are limited. Xie and Qin [10] proposed a pruned lite ML model for distributed semantic coders. Their proposed method is a learning model for trained learning models rather than for coder updating. Furthermore, Shi et al. [11] and Qin et al. [12] suggested general FL frameworks for semantic coder updating in networks. Nonetheless, these frameworks incur long service interruptions, energy consumption, and privacy risks in SEC networks. The above research works highlight the importance of designing efficient collaborative learning methods for updating the semantic coders in 6G networks is essential.

Several studies investigate the employment of SemCom for AEC devices. Kang et. al [31] proposed a new aerial semantic image transmission paradigm based on deep reinforcement learning (DRL) to improve the transmission accuracy of UAVs. In [32], semantic communication was integrated into their presented DRL framework for increasing communication reliability and decreasing the latency of air-terrestrial networks. Kang et. al [33] introduced a task-oriented semantic communication framework for UAVs. The UAV sends only the necessary images to the required users rather than all images, thus reducing its energy consumption. Nevertheless, these existing studies for semantic communication much more concentrate on AEC devices but neglect to take into account the influence of SemCom in AENs.

In SAT networks, adopting SEC for users in remote areas or disaster zones has been recently investigated in [34] and [35]. The authors in [34], and [35] mainly focused on developing offloading decisions that minimise offloading delay or energy consumption for cases where users have direct radio links to the satellites. (e.g., in C-Band). An alternative access scenario is proposed in [36], where the user transmits to the SEC indirectly through an intermediary TST. In this approach, the user transmission to the TST is on a C-band radio link and TST communicates to the SEC through a K-band radio link. Wang et al. [37] also proposed a dual-edge cloud network, where the edge servers are placed on both BSs and LEO satellites. In this approach, a BS acts as a TST to assist users with computation offloading to the SEC. Similarly, [38] proposed an energy-efficient strategy for terrestrial users to offload computing tasks to the SEC via TSTs. Tang et al. [39] further investigated the impact of the core cloud on users' offloading decisions. They then proposed a minimal energy consumption computing offloading decision method, where users access SEC directly. The above approaches often limit their investigations to one connectivity scenario between the users and the SEC, while considering only part of the performance (e.g., energy or latency) and overlooking the potential privacy issues associated with offloading users' tasks elsewhere. Further, SemCom was also not integrated.

A few resource-optimal studies have also been proposed for SemCom-assisted networks. Yan et. al [40] defined the semantic spectral efficiency optimisation for resource allocation in terms of channel assignment and the number of transmitted semantic symbols. In [41], compression ratio and resource allocation were optimised jointly to maximize the success probability of tasks. Furthermore, quality-of-experience aware resource allocation in terms of the number of transmitted semantic symbols, channel assignment, and power allocation was introduced in [42]. However, these allocation strategies focus more on communication cost than on computation cost. In addition, they also ignore the privacy, the resource variations associated with online training of semantic coders and the differences in the specific application scenarios of SemCom.

2.5 Summaries

Therefore, there are extremely limited studies on integrating SemCom systems in 6G networks. In addition, SemCom deployment in networks faces the problem of semantic coder updates and the new problematic resource allocation concerns it entails that also urgently need to be resolved. In this thesis, SemCom systems for various potential 6G 3D networks will hence proposed and developed. Moreover, SemCom coder updating mechanisms and resource allocation schemes will presented for proposed SemCom systems with comprehensive consideration of communications and computing costs, as well as potential privacy risks.

Chapter 3

Terrestrial Vehicular SemCom System

3.1 Introduction

In this chapter, the challenges of SemCom in terrestrial vehicular networks are analysed and investigated. How to efficiently update semantic coders in the network in real-time is amongst the main challenges for the SemCom system design. Nevertheless, the existing studies are extremely limited in addressing this challenge of SemCom as mentioned in Chapter 2. I can summarise the deployment of the existing frameworks for SemCom, i.e., frameworks based on FL, in vehicular SemCom networks for task offloading faces the following challenging questions:

Q1: Encoders that extract semantic information from different vehicles may have different models. This prevents the vehicle from participating in coder model aggregation for FL.

Q2: FL requires the entire coder (encoder and decoder) to be trained on the vehicle. This however significantly increases the computational workload on the vehicle. In addition, the required storage of the trained decoder model for each type of transmission content increases the vehicle's storage overhead.

Q3: The high mobility of vehicles also presents the challenge of selecting appropriate vehicles for collaborative training. There is also a trade-off to be made in terms of technical factors such as training delays, and energy costs.

In this chapter, I provide tractable solutions to these questions. Split learning (SL) [43] is a new distributed training approach proposed in the ML domain recently. However, it is also not applicable to vehicular semantic networks. The loss value required for trained coder updating is unavailable in a dynamic vehicle environment due to the SL splitting the training model to be trained on different devices. In this chapter, I show that combining the advantages of FL with SL is a potential scheme for semantic coder updating in mobile vehicular networks. I propose a mobility-aware split-federated learning framework to address these urgent needs for considered vehicular SemCom networks. A TL [13] paradigm for vehicular SemCom is then proposed to be integrated into the presented framework by employing part of the trained encoder. I refer to our proposed mobility-aware split-federated with transfer learning framework as MSFTL. Moreover, a high-mobility training energy optimisation mechanism for MSFTL is also presented based on the Stackelberg game.

The rest of this chapter is organised as follows: Section 3.2 presents the vehicle SemCom system model. The proposed MSFTL framework and the analysis of its computing and communication overhead are presented in Section 3.3. In Section 3.4, the game theoretical mechanism design is proposed for resource optimisation. Section 3.5 presents the simulation results showing that our proposed framework and mechanism achieve excellent performance. Finally, this chapter is concluded in Section 3.6.

3.2 System model

In this section, I first introduce the vehicular SemCom network traffic model, and then the vehicle computational and communication workload models are presented.

3.2.1 Vehicle SemCom model

In this chapter, I assume a set of TECs, $\{1, 2, ..., m, ..., M\}$, is deployed on roadside units (RSUs) or base stations (BSs) and a set of vehicles $\{1, 2, ..., n, ..., N_m\}$ is in the service range of TEC m (Figure 3.1). Further, there are I_m vehicles in TEC m's range that participate in the DLJSC coder model training. Different vehicles transmit the offloading content via various models of DLJSC encoder to the TEC, where the TEC receives it via a DLJSC decoder. When the vehicle or TEC semantic knowledge base is scarce, vehicles need to be selected for participation in the training based on the vehicle‘s velocity. According to [44], I can have the average velocity (km/h) $\bar{v}_m$ of N_m vehicles in the service range of TEC m as:

$$\bar{v}_m = \max\{v_{m_{max}} = (1 - \frac{N_m}{N_{m_{max}}}), v_{m_{min}}\}, \tag{3.1}$$

where $v_{m_{max}}$ is the maximum vehicle velocity that can be driven within the service range of TEC m. I assume roads in the TEC service range are uniform and have the same permissible maximum vehicle velocity. Similarly, $v_{m_{min}}$ is the vehicle velocity when the road is congested. Further, $N_{m_{max}}$ is the maximum allowable number of vehicles in TEC m's service range on the road. In the case of free-flow traffic conditions, the velocity of a vehicle n in the service range of TEC m, $v_{n,m}$ is a normally distributed random variable with the probability density function given by [44]

$$f(v_{n,m}) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(v_{n,m} - \bar{v}_m)}{2\sigma^2}}, \tag{3.2}$$

where $\sigma = k\bar{v}_m$ and $v_{m_{min}} = \bar{v}_m - l\bar{v}_m$. The two-tuple (k, l) is subject to the traffic activity observed in real-time. I can also rewrite it as:

$$\hat{f}(v_{n,m}) = \frac{f(v_{n,m})}{\int_{v_{m_{min}}}^{v_{m_{max}}} f(v_{n,m}) dv_{n,m}} = \frac{2f(v_{n,m})}{\text{erf}(\frac{v_{m_{max}} - \bar{v}_m}{\sqrt{2}\sigma}) - \text{erf}(\frac{v_{m_{min}} - \bar{v}_m}{\sqrt{2}\sigma})}. \tag{3.3}$$

3.2.2 Computing and communication model

I consider a vehicle computing offloading scenario, where vehicle n in the service range of TEC m has a task with data size $k_{n,m}$ to offload. Further, I assume the

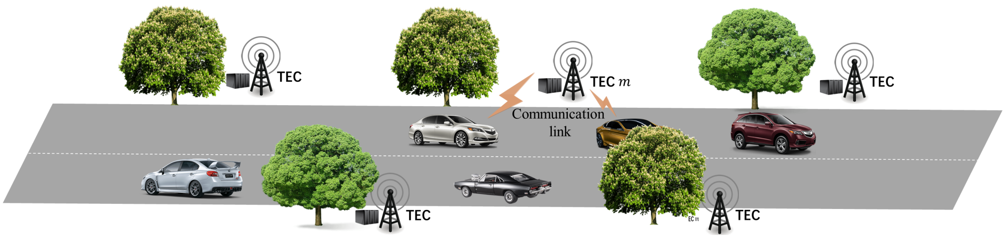


Figure 3.1: Vehicles in the network.

size of training data to be computed by this vehicle during coder model training is $d_{n,m}$. I write the training delay of one epoch as:

$$T_{n,m} = \frac{d_{n,m}}{f_{n,m}}, \tag{3.4}$$

where $f_{n,m}$ is the CPU-cycle frequency of vehicle n with the unit cycles/s. The energy cost is [38]

$$E_{n,m} = p^c_{n,m} T_{n,m} = \varepsilon f^3_{n,m} \frac{d_{n,m}}{f_{n,m}} = \varepsilon d_{n,m} f^2_{n,m}, \tag{3.5}$$

where ε is the energy parameter depending on chip [45] and $p^c_{n,m}$ is computing power.

According to the Shannon theory, the communication delay for transmitting a task $k_{n,m}$ should be

$$t_{n,m} = \frac{k_{n,m}}{r_{n,m}} = \frac{k_{n,m}}{B_{n,m} \log_2(1 + \frac{p_{n,m} g_{n,m}}{\sigma_0^2})}, \tag{3.6}$$

where $r_{n,m}$ is the transmission rate. Further, $B_{n,m}$ is the bandwidth, $p_{n,m}$ is transmission power and $g_{n,m}$ is the channel gain. Thus, the transmission energy cost is

$$e_{n,m} = p_{n,m} t_{n,m}. \tag{3.7}$$

SemCom differs from traditional communication in spectral efficiency research [40],[46]. Conventional communications focus on unit bandwidth rates, while SemComs focus on effective semantic information delivered per second. I also consider that in practical signal transmission, the transmission process of SemCom is still based on traditional communication theory as described above.

3.3 MSFTL for vehicle SemCom

In this section, the new TL-based approach for the vehicle network QoS enhancement is presented. I also present the details of our proposed MSFTL framework. Finally, I compare the computational and communication cost of the proposed MSFTL framework with that of the conventional FL framework.

3.3.1 Transfer learning for vehicle SemCom network

The successful application of Autoencoder, a deep unsupervised learning model, has recently been demonstrated in the design of SemCom architectures [16], [47], [48]. It extracts the input features by downscaling features via the encoder and subsequently the image is recovered through the decoder. The autoencoder training process entails converting inputs, $\boldsymbol{x}$, into intermediate feature variables $\boldsymbol{y}$ via the encoder part. Therefore, variables, $\boldsymbol{y}$, are converted into $\tilde{\boldsymbol{x}}$ by the decoder part. Finally, inputs $\boldsymbol{x}$ and outputs $\tilde{\boldsymbol{x}}$ are compared to ensure that they are both infinitely close. Nevertheless, training from scratch often takes a long time and a significant number of samples. Depending on the network composition of the autoencoder, such as based on transformer [20] or convolutional neural network (CNN) [15], the training time varies.

To address these challenges, I propose a TL approach. In this approach, I develop the un-updated DLJSC encoder model in two parts: the pre-training model, and fine-tuning layers. Every vehicle allows having various types of the pre-training model. The pre-training model is a part of the encoder model which is the vehicle encoder that has been trained over a long period of time with a large amount of data. However, this model is not well suited to the required training task of feature extraction. Hence, in our model, the last layers of the vehicle semantic encoder are replaced with the same type of untrained layers. The replaced layers are called fine-tuning layers which are trained for a specific task. The vehicle does not need to retrain the pre-trained model again. Only the last few layers of the encoder need to be trained. Furthermore, to alleviate the small sample size issues, fine-tuning layers are trained together at the edges, as specified below. Therefore, vehicles only need to ensure the last few layers of the encoder have the same model. The storage resource required and training costs for different missions are thus reduced.

3.3.2 MSFTL design

Considering the pervasive case of semantic coders update, I propose a novel training framework based on split-federated learning for vehicle SemCom networks. SL is a collaborative learning approach in distributed systems designed to learn models for clients [43]. SL splits the model into two parts, one on the decentralised clients and another one on a centralised server. Multiple clients jointly train a shared model on the centralised server together with their part of the model. Therefore, it enables data information sharing and reduces the computational load. FL [30],[49] is also a distributed collaborative learning approach, where clients train the entire model and finally aggregate the model weight on the server. Thus protecting privacy and enabling the indirect sharing of data. The aggregation method generally employs the widely adopted Federated Averaging (FedAVG) algorithm [50],[51]. It is based on the weighted average for weight aggregation.

Nevertheless, SL is not very suitable for training server models as the calculation of loss values requires private raw data that is not available at the same place as the loss value calculation. Further, FL requires identical models for federated aggregation which means FL require the same encoder model in our considered vehicular semantic networks. Therefore, based on the above, neither of these traditional frameworks can be applied to the vehicle SemCom network as they face the *Q1*-*Q3* and privacy challenges. In our proposed MSFTL (Figure 3.2), the advantages of both SL and FL are sustained, while the mentioned challenges are also tackled. The coder is split into four parts during training, including the pre-training model P_1, the fine-tuning layers P_2, the TEC private decoder (part of the decoder) P_3 and the last layer of the decoder P_4. The entire model is split but

Algorithm 1 MSFTL for vehicular semantic communication

```
After confirming trainable vehicles
Vehicle Execution:
Batch size: $J$
for each local epoch $a = 1, 2, ...A$
  From EC $m$ get $P_4^m$ weight parameters $W_4^{a-1}$
  for each vehicle involved in training $n = 1, 2, ...I$
    From EC $m$ get $P_3^m$ forward propagation output $\tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{n}}$
    for each local batch $b_n = 1, 2, ...$
      Forward propagation in $P_4^m$ and get output $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{i}}$
      Loss $y \longleftarrow \frac{1}{J}\sum_{j=1}^{J}(x_j^n - \tilde{\boldsymbol{x}}_{\boldsymbol{4,j}}^{\boldsymbol{n}})$
      Get backpropagation output $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{i'}}$ and send back $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{i'}}$
      Update $W_{4,n}^a$
    end for
      Transmit $W_{4,n}^a$ to EC $m$
  end for
end for
EC $m$ Execution:
From each vehicle $n$ involved in training get $P_1^{n,m}$ output $\tilde{\boldsymbol{x}}_{\boldsymbol{1}}^{\boldsymbol{n}}$
  for each epoch $a = 1, 2, ..., A$
    Forward propagation in $P_2^m$ and $P_3^m$, and get output $\tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{n}}$ for each vehicle $n$
    After vehicles training ...
    Get $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{n'}}$ from vehicles and perform backpropagation
    Update $W_2^a$ & $W_3^a$
    Get $W_{4,n}^a$ from vehicles
    Update $W_{4,n}^a$
  end for
```

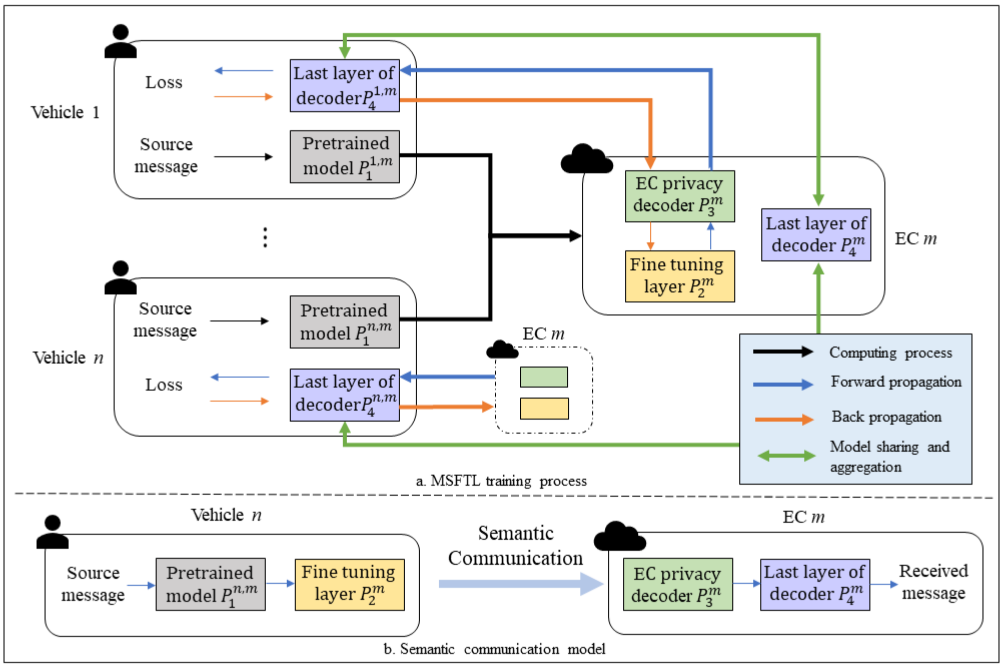

Vehicle 1
Loss
Last layer of decoder$P_4^{1,m}$
Source message
Pretrained model $P_1^{1,m}$
Vehicle n
Source message
Pretrained model $P_1^{n,m}$
Loss
Last layer of decoder$P_4^{n,m}$
EC privacy decoder P_3^m
Fine tuning layer P_2^m
Last layer of decoder P_4^m
EC m
EC m
Computing process
Forward propagation
Back propagation
Model sharing and aggregation
a. MSFTL training process
Vehicle n
Source message
Pretrained model $P_1^{n,m}$
Fine tuning layer P_2^m
Semantic Communication
EC m
EC privacy decoder P_3^m
Last layer of decoder P_4^m
Received message
b. Semantic communication model

Figure 3.2: The functional block diagram of the proposed MSFTL.

trained together. Trainable vehicles are selected based on factors such as velocity, and computational capability. I will elaborate on the details in the next section.

The SemCom model update algorithm is shown in Algorithm 3.1. Firstly, the trainable vehicles and training data are identified. These are based on the Stackelberg game based resource optimisation mechanism. We will elaborate on the details in the next section.

In the coders' training process, the pre-training model, P_1, and the last layer of the decoder, P_4, are trained on the vehicle while fine-tuning layers P_2 and the EC private decoder P_3 are trained on the EC.

For a trainable vehicle n in EC m's range, the fuzzy features $\tilde{\boldsymbol{x}}_1^{n,m}$ are first extracted from training samples $\boldsymbol{x}^{n,m}$. The features $\tilde{\boldsymbol{x}}_1^{n,m}$ are obtained through a freezing pre-training model $P_1^{n,m}$ and transmitted to the EC m. Subsequently, the EC m treats fuzzy features $\tilde{\boldsymbol{x}}_1^{n,m}$ as inputs and start the training cycle. In one epoch, the EC uses $\tilde{\boldsymbol{x}}_1^{n,m}$ performing forward propagation training of the fine-tuning layer P_2^m and the EC private decoder P_3^m. The results of the forward propagation from

P_3^m, i.e., $\tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{n,m}}$, are sent to the corresponding vehicle n. The corresponding vehicle n then trains the last layer of decoder $P_4^{n,m}$ and gets output $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{n,m}}$. Thereafter, the vehicle gets the loss value $L^{n,m}$ by comparing the variability between source message $\boldsymbol{x}^{\boldsymbol{n,m}}$ and forward propagation output $\tilde{\boldsymbol{x}}_{\boldsymbol{4}}^{\boldsymbol{n,m}}$. The backpropagation process is then carried out based on $L^{n,m}$ and returning along with the same path until fine-tuning layers P_2^m. Finally, since the last layer of the encoder $P_4^{n,m}$ has only been trained for a single vehicle, a federated aggregation is required to guarantee that the decoders are identical.

The vehicles participating in the training send it to EC m for aggregation, which then returns the aggregation result P_4^m to each sending vehicle. All vehicles involved in the training complete a training epoch after performing the process once. After the training, $P_1^{n,m}$ and P_2^m forms the vehicle n's DLJSC encoder. Similarly, P_3^m and P_4^m forms the EC's DLJSC decoder. During the whole process, the user's private information $P_1^{n,m}$ and $\boldsymbol{x}^{\boldsymbol{n,m}}$ is not leaked, i.e., the client encoder models can be different, and the privacy of clients is protected. The vehicle only needs to replace the fine-tuning layer for different transmission contents, thus reducing the vehicle's storage load.

3.3.3 Comparison of computing and communication overhead

For vehicles, regardless of the employed collaborative learning framework, a certain degree of computational and communication load is expected. Neither FL nor SL is applicable to the vehicle SemCom network due to the *Q1-Q3* and privacy challenges. However, to enable making the employment of FL, I can assume that the vehicle encoder models are the same. To further validate the advantages of our MSFTL in the following, I compare the computational and communication load of the existing FL framework with the proposed MSFTL for the same encoder model.

I assume the total number of training epochs is e. The computational delay of the vehicle n in the service range of TEC m to be consumed by the model update in the FL framework is expressed as:

$$T_{n,m}^{FL} = D_{n,m}\frac{d_P}{f_{n,m}}e, \tag{3.8}$$

where d_P is the size of the computation required for the coder model of one training data in one epoch and $D_{n,m}$ is the number of training data from vehicle n. Therefore, the required energy for computations is

$$E_{n,m}^{FL} = \epsilon D_{n,m} d_P f_{n,m}^2 e. \tag{3.9}$$

In contrast to FL, the imposed computational delay and energy of the proposed MSFTL can be expressed as:

$$T_{n,m}^{MSFTL} = D_{n,m}(\frac{d_{P_4^{n,m}}}{f_{n,m}}e + \frac{d_{P_1^{n,m}}}{f_{n,m}}), \tag{3.10}$$

$$E_{n,m}^{MSFTL} = \epsilon D_{n,m}(d_{P_4^{n,m}} f_{n,m}^2 e + d_{P_1^{n,m}} f_{n,m}^2), \tag{3.11}$$

where $d_{P_1^{n,m}}$ is the size of the computation needed to derive the output $\tilde{x}_1^{n,m}$ from the pre-trained model. Furthermore, $d_{P_4^{n,m}}$ is the training computation load of the final layer of the decoder. Hence, for the same coder model,

$$d_P > d_{P_1^{n,m}} + d_{P_4^{n,m}}. \tag{3.12}$$

I can also write:

$$T_{n,m}^{FL} > T_{n,m}^{MSFTL}, \tag{3.13}$$

$$E_{n,m}^{FL} > E_{n,m}^{MSFTL}. \tag{3.14}$$

Therefore, our proposed framework requires a lower computational cost in vehicles than FL.

I express the communication cost during training in terms of communication rounds for visual representation. FL requires clients to offload the trained model weights to the TEC and return them after TEC aggregation in each training epoch. FL therefore communication load of vehicle n is

$$C_{n,m}^{FL} = 2\omega_p e, \tag{3.15}$$

where ω_p is the size of coder model weights. Therefore, the communication cost of federated the last layer of the decoder is

$$C1_{n,m}^{MSFTL} = 2\omega_{p_4^{n,m}} e, \tag{3.16}$$

where $\omega_{p_4^{n,m}}$ is the size of the last layer of the decoder weights. As MSFTL requires the client to first send the pre-trained model output $\tilde{x}_1^{n,m}$ to the TEC, the TEC and client need to perform forward and backpropagation of the final layer of the decoder. The split training communication load is therefore

$$C2_{n,m}^{MSFTL} = O_1^{n,m} D_{n,m} + 2O_3^{n,m} D_{n,m} ep, \tag{3.17}$$

where $O_1^{n,m}$ and $O_3^{n,m}$ are the number of output layer neurons of pre-trained model $P_1^{n,m}$ and partial decoder model P_4^m, respectively. Thus, the total communication load of the proposed MSFTL is

$$C_{n,m}^{MSFTL} = C1_{n,m}^{MSFTL} + C2_{n,m}^{MSFTL} = 2e(\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}) + O_1^{n,m} D_{n,m}. \tag{3.18}$$

Since e is usually a large number, I have $2e(\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}) >> O_1^{n,m} D_{n,m}$. Therefore, I ignore $O_1^{n,m} D_{n,m}$ in the comparison. Hence, the comparison of the communication cost of the FL and MSFTL can be expressed as ω_p versus $\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}$. I can conclude that MSFTL is more communication efficient in case the amount of the coder model weight is larger, otherwise, FL performs better. Nevertheless, FL only applies to special cases where the encoders of all vehicle models are the same. In contrast, our proposed MSFTL not only adapts to variable network environments but also performs better in terms of computational load.

3.4 Stackelberg game based resource optimisation mechanism

In this section, I present a high-mobility training energy optimisation mechanism for the MSFTL. The mechanism is based on the Stackelberg game, which jointly takes into account vehicle mobility and minimises training energy costs. First, I present the game at vehicles in the mechanism and the selection of training vehicles considering mobility. I then introduce the design of the game at the TEC and present mechanism optimisation formulation and its solution.

3.4.1 Game design at the vehicles

It is important to ensure that the vehicle has sufficient training time before training. First, I analyse the available training time for the vehicle. I assume $D_{n,m}$ is the number of training data participants training from vehicle n in the range of TEC m and $D_{n,m}^{max}$ is the maximum available training data from vehicle n. Further, I assume that the communication status of vehicle n remains constant during training. The duration of the training can be expressed as:

$$\Psi_{n,m} = D_{n,m}(\frac{d_{P_4^{n,m}}}{f_{n,m}}e + \frac{d_{P_1^{n,m}}}{f_{n,m}} + \frac{zO_1^{n,m} + 2zO_3^{n,m}e}{B_{n,m}\log_2(1 + \frac{p_{n,m}g_{n,m}}{\sigma_0^2})}) + \textstyle\sum_n^{I_m} D_{n,m}(\frac{d_{P_{2,3}^m} + d_{P_4^m}}{f_m})e, \tag{3.19}$$

where z is the parameter to convert the data number to the size to be transmitted and f_m is the CPU-cycle frequency of TEC m. Further, $d_{P_{2,3}^m}$ is the training computation size of P_2^m and P_3^m, and $d_{P_4^m}$ is federated aggregation computation load. Moreover, I_m is the number of trainable vehicles and $\sum_n^{I_m} D_{n,m}$ denotes the total number of training data submitted from the trainable vehicles. For simplicity, I set

$$\Psi_{n,m} = D_{n,m}\Gamma_{n,m} + \sum_n^{I_m} D_{n,m}(\frac{d_{P_{2,3}^m} + d_{P_4^m}}{f_m})e. \tag{3.20}$$

The vehicle residence time can be estimated as:

$$K_{n,m} = \frac{h_{n,m}}{\bar{v}_m}, \tag{3.21}$$

where $h_{n,m}$ is the distance that the vehicle n travels out of the TEC m's service range. Moreover, $\bar{v}_m$ is the vehicles' average velocity in TEC m's service range mentioned in Section 3.2. To ensure learning efficiency, $h_{n,m}$is considered as the shortest distance at multiple forks in the road. Therefore, trainable vehicles should satisfy $\Psi_{n,m} \leq K_{n,m}$, that is

$$D_{n,m}^{min} \leq D_{n,m} \leq D_{n,m} \frac{K_{n,m}}{\Psi_{n,m}}, \tag{3.22}$$

where $D_{n,m}^{min}$ is the minimum training data required to guarantee accuracy.

Once suitable trainable vehicles have been identified, semantic coder model training can be initiated. I mainly consider the computational and communication energy cost of the vehicle during training. Energy cost is defined as cost and the cost of vehicle n can be denoted by

$$\Theta_{n,m} = E_{n,m}^{MSFTL} + D_{n,m} \frac{z p_{n,m} O_1^{n,m} + 2 z p_{n,m} O_3^{n,m} e}{B_{n,m} \log_2(1 + \frac{p_{n,m} g_{n,m}}{\sigma_0^2})}. \tag{3.23}$$

Nevertheless, the vehicle is not necessarily willing to participate in the training due to the different situations faced. Sufficient data is one of the guarantees of model accuracy. I hence set a pricing function and design a game for the vehicles to incentivise the vehicles to participate in the training. To ensure fair allocation of bonuses, I use a weight-sharing model commonly used in the game bonuses design. I write

$$R_{n,m} = \frac{\omega_{n,m} D_{n,m}}{\sum_{n}^{I_m} \omega_{n,m} D_{n,m}} R_m, \tag{3.24}$$

where R_m is the total bonus from the TEC and $\omega_{n,m}$ is the coefficient depending on the quality of vehicle communication as it affects the quality of transmitted data. Here, $R_{n,m}$ and R_m have no unit, they are numerical values and they are judged by comparing the magnitudes. The corresponding coefficient of vehicle n is $\omega_{n,m}$. Hence, I have the utility function of the game at vehicles as:

$$\mu_{n,m} = \alpha R_{n,m} - \beta \Theta_{n,m}, \tag{3.25}$$

where α and β are normalisation factors enable $\alpha R_{n,m} \leq 1$ and $\beta \Theta_{n,m} \leq 1$. This allows the utility function to be a pure numerical function and the utility value is a unitless number. I can further define the vehicles' game problem as:

Problem 3.1:

$$\max_{D_{n,m}} \quad \alpha R_{n,m} - \beta \Theta_{n,m}, \tag{3.26a}$$

$$s.t. \quad D_{n,m} \geq D_{n,m}^{min}, \tag{3.26b}$$

$$D_{n,m} \leq D_{n,m} \frac{K_{n,m}}{\Phi_{n,m}}. \tag{3.26c}$$

3.4.2 Game design at the TEC

In this subsection, I design the game at the TEC and its utility function. I assume the accuracy of the model is related to the amount of training data. The objective of the TEC is to minimise the reward offered while satisfying the minimum QoS (accuracy) after training. Without loss of generality, the TEC m's utility is defined as:

$$U_m \triangleq \gamma \Omega - \delta R_m, \tag{3.1}$$

where γ and δ are normalisation factors and Ω is a function related to the accuracy of the training model. As the relationship between the amount of training data and the accuracy of the model shows an increasing trend with a gradual decrease in the rate of growth in our simulation (Figure 3.7). I thus use a logarithmic function to model the Ω as:

$$\Omega \triangleq \ln(1 + \theta \sum_{n}^{I_m} D_{n,m}), \tag{3.2}$$

where θ is a parameter related to the training model. Further, it is limited to more than minimum permissible the accuracy Ω^{min} and less than the maximum accuracy Ω^{max} possible for the model. The game problem at the TEC thus can be written as:

Problem 3.2:

$$\max_{R_m} \quad \gamma \ln(1 + \theta \sum_{n}^{I_m} D_{n,m}) - \delta R_m, \tag{3.3a}$$

$$s.t. \quad R_m > 0, \tag{3.3b}$$

$$\Omega^{min} < \ln(1 + \theta \sum_{n}^{I_m} D_{n,m}) \leq \Omega^{max}. \tag{3.3c}$$

3.4.3 Optimal solutions and equilibrium analysis

NE Existence: Problem 3.1 (follower) and Problem 3.2 (leader) form a Stackelberg game. I assume $D^*_{n,m}$ and R^*_m are the optimal solutions for Problem 3.1, and Problem 3.2, respectively. Thus, the game needs to satisfy the following equation to reach Stackelberg Equilibrium (SE) point(s)

$$\mu(D^*_{n,m}, R^*_m) \geq \mu(D_{n,m}, R^*_m), \tag{3.1}$$

$$U(D^*_{n,m}, R^*_m) \geq U(D^*_{n,m}, R_m). \tag{3.2}$$

It is found from Problem 3.1 that the strategy set at vehicles is compact and convex. Further, as the second order partial derivative is less than zero, i.e., $\frac{\partial^2 \mu_{u,m}}{\partial D^2_{n,m}} = -\frac{2\omega^2_{n,m} R_m \sum^{I_m}_{j,j\neq n} \omega_{j,m} D_{j,m}}{(\sum^{I_m}_{j,j\neq n} \omega_{j,m} D_{j,m} + D_{n,m}\omega_{n,m})^3} < 0$, the utility function is continuous and concave in $D_{n,m}$. Thus, according to the Debreu-Glicksberg-Fan theorem a pure NE exists [52].

I then employ classic backward induction to find SE points. The optimal strategies for vehicles are obtained first, followed by the optimal strategy for the TEC. If the vehicle residence time is less than the minimum trainable time, i.e., $K_{n,m} < \Psi_{n,m}(D^{min}_{n,m})$. Then $D_{n,m}* = 0$. If $K_{n,m} \geq \Psi_{n,m}(D^{min}_{n,m})$, by deriving the first order partial derivative of (3.26a) with respect to $D_{n,m}$, I have

$$\frac{\partial \mu_{n,m}}{\partial D_{n,m}} = \alpha \frac{\omega_n \sum^{I_m}_{j,j\neq n} \omega_{j,m} D_{j,m}}{(\sum^{I_m}_{n} \omega_{n,m} D_{n,m})^2} R_m - \frac{\beta \Theta_{n,m}}{D_{n,m}}. \tag{3.3}$$

For simplicity of presentation, I set $H_{n,m} = \frac{\beta\Theta_{n,m}}{D_{n,m}}$. In case that (32) equals 0, the optimal training data obtained as $f_{n,m}(D^*_{n,m} R_m) = \sqrt{\frac{\alpha R \sum^{I_m}_{j,j\neq n} \omega_{j,m} D_{j,m}}{\omega_{n,m} H_{n,m}}} - \frac{\sum^{I_m}_{j,j\neq n} \omega_{j,m} D_{j,m}}{\omega_{n,m}}$ and the TEC's utility function can be written as:

$$U_m = \gamma ln(1 + \theta \sum^{I_m}_{n} f_{n,m}(D^*_{n,m}, R_m)) - \delta R_m. \tag{3.4}$$

Due to the high complexity and multiple constraints, sub-games NE cannot be derived in a closed form. Therefore, I solve the game in two segments through numerical search. In the first step, I employ the simplicial method [53] to achieve

Algorithm 3.2 Stackelberg game-based energy optimisation mechanism

```
Set the maximum number of iterations $K$, and learning rate $\theta$
Set initial positive numbers for $R$ and $D_i$
**while** $k < K$
    $D_i(k) \longleftarrow \sqrt{\frac{\alpha R \sum_{j,j\neq n}^{I_m} \omega_j D_j}{\omega_i H_i}} - \frac{\sum_{j,j\neq n}^{I_m} \omega_j D_j}{\omega_i}$
    $D_i^*(k) \longleftarrow$ constraints and $D_i$
    $U(k) \longleftarrow R(k)$ and $D_i^*(k)$
    $R(k+1) = R(k) + \theta$
**end while**
Find the maximum $U(k)$ and corresponding $R(k)$ and $D_i^*(k)$
**return** $R(k)$ and $D_i^*(k)$
```

each $D_{n,m}$‘s optimal decision by solving a piecewise linear approximation of the problem while holding R_m fixed. Subsequently, $f_{n,m}(D_{n,m}, R_m)$ is substituted in (3.33), R_m is updated using the two-dimension grid search, and R_m is substituted back into the first step. $D_{n,m}$ and R_m thus iteratively tighten until convergence. The solution algorithm is shown in Algorithm 3.2.

3.4.4 MSFTL

3.5 Simulation results

In this section, I evaluate the performance of the proposed MSFTL and optimisation mechanism. First, I compare the proposed MSFTL framework with the existing FL framework for SemComs in terms of convergence speed, and accuracy. Then, the advantage of the presented optimisation mechanism based on the Stackelberg game is assessed in a variety of different scenarios.

I first elaborate on the simulation settings in evaluating the performance of our proposed framework and ignore the communication noise when training. The

Table 3.1: The setting of the CAE in the proposed semantic network framework.

	LayerName	**Number of neurons**
Pre-training model	Conv+ReLU	128
	Conv+Pool+ReLU	64
	Conv+Pool+ReLU	32
Fine-tuning layer	Conv+Sigmoid	10
TEC private decoder	transConv+ ReLU	10
	transConv+ ReLU	32
	transConv+ ReLU	64
Final layer of decoder	transConv+ Sigmoid	128

adopted SemCom model is based on convolutional autoencoder (CAE) [15], the details of the CAE setting are shown in Table 3.1. Since the baseline frameworks for SemCom networks are limited and all based on FL, e.g., [30],[49], to enable the FL to operate in a vehicle SemCom network, I assume all users have the same encoder model and the same degree of pre-training. Further, training and pre-training datasets employed are CIFAR 10 and CIFAR 100 [54], respectively. They are both composed of a 50,000-image training set and a 10,000-image test set. The difference is that CIFAR 10 has 10 classes, while CIFAR 100 has 100 classes.

In order to more realistically verify the performance of the proposed framework in the case of vehicle task offloading, I set the experimental environment to object/image recognition after computing offloading. I validate the classification of the transmitted images using a fully trained VGG16 [55] network, and its accuracy comparison with the images before transmission visualizes the performance of the frameworks. I also assume the similarity of the recognition accuracy of the object/image after transmission in VGG16 compared to before transmission as the SemCom model accuracy. In addition, the number of users involved in the training of our network is 10 and the sample set is divided randomly and equally into 10 copies, if not stated in particular.

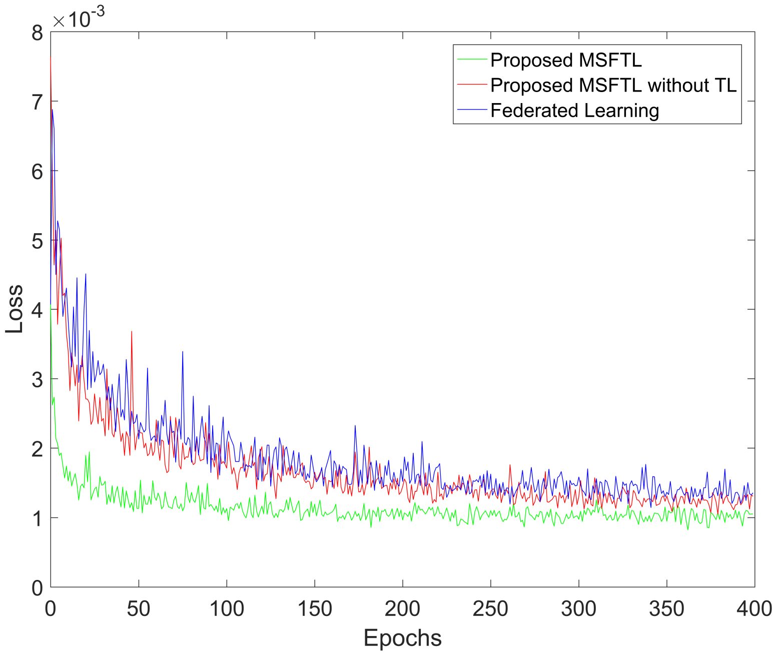


Figure 3.3: Convergence speed comparison of different frameworks.

Figure 3.3 illustrates the performance of the proposed MSFTL in terms of convergence speed. I set the batch size as 64 and compared the proposed MSFTL with the FL framework and the MSFTL without the TL model. I can observe that as the number of training times increases, the loss values of each approach gradually decrease and eventually plateau. The decrease curve of the MSFTL without TL almost coincides with FL, proving that both sides can achieve almost similar performance in terms of convergence. Nevertheless, our proposed MSFTL convergence rate and the final loss values achieve a very significant outperformance. This is because the pre-training model accelerates the training and a well improves the model feature extraction capability.

Figure 3.4 presents the image offloading accuracy of CAEs trained by different training frameworks for different numbers of participating vehicles. It can be seen that the accuracy of all the training frameworks increases as the number of participating vehicles increases. This is because the increase in the number of participating vehicles leads to an increase in the total training sample. Furthermore,

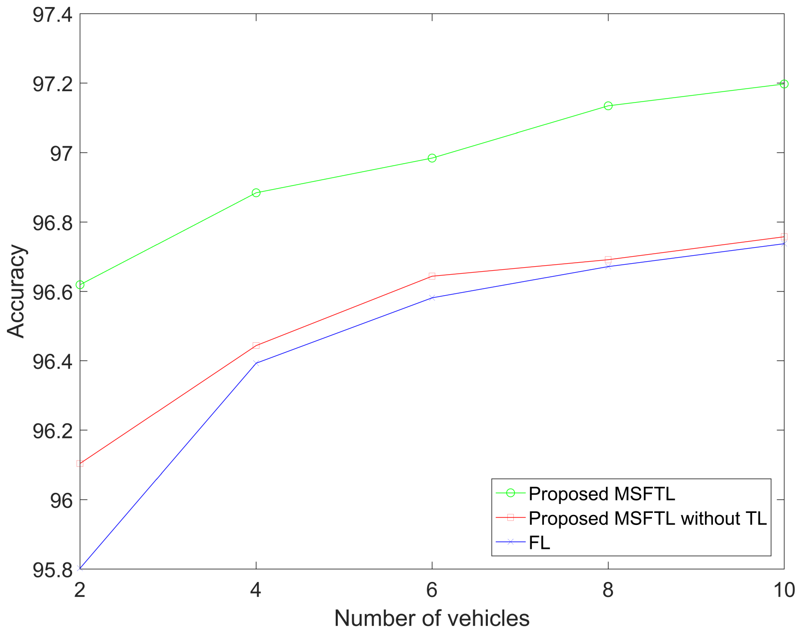


Figure 3.4: Accuracy of different frameworks.

our proposed MSFTL consistently achieves the optimal transmission/offloading accuracy as varying numbers of vehicles are involved in the training. This increases the QoS of vehicle task offloading. Moreover, although the accuracy is not smoothly increasing as the number of vehicles (samples) increases due to the stochastic property of machine learning, it is still noticeable that the trend is similar to the log function. It validates Eq. (3.28) in our game design.

Figure 3.5 shows the computing cost of the vehicle under different training frameworks. For comparison purposes, I define the computing cost as the number of neurons that need to be computed in the forward and backpropagation of the vehicle in one Epoch. Vehicles are not limited to aggregating only the last layer of the encoder. Furthermore, FL is set to a constant value due to its aggregation of all weights. It can be observed that the vehicle computing cost increases as the number of layers to be aggregated increases. When all the last five layers need to be aggregated, it has the same computing cost as FL. This is because all the network models are trained on the vehicles at that moment. Our proposed MSFTL reduces

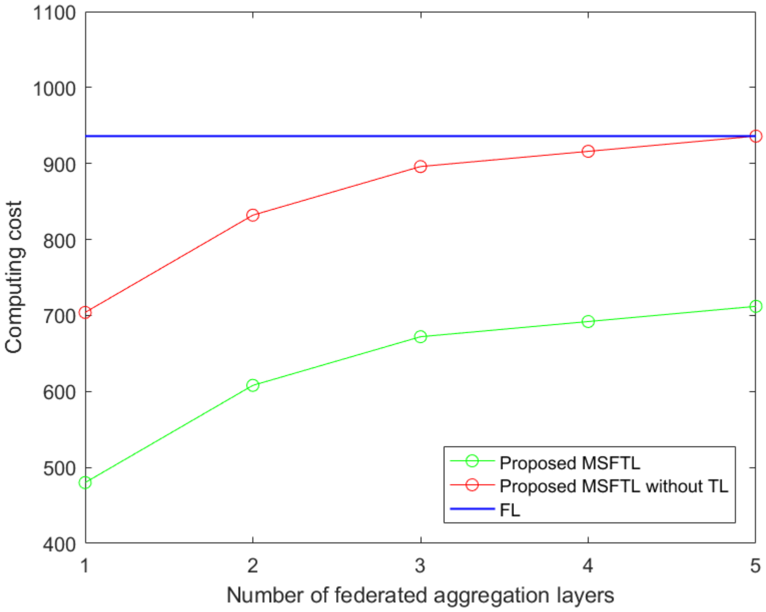


Figure 3.5: Computing cost of different frameworks.

the backpropagation overhead of the pre-training model due to the presence of TL so that the vehicle computing cost is always kept at the lowest of all frameworks. Further, the aggregation of the last layer decreases the computing cost for the vehicle and simultaneously mitigates the risk of model privacy leakage.

Figure 3.6 evaluates the communication cost of the different frameworks in one Epoch. As the analysis in Session III-C, our proposed framework communication cost involves the federated aggregation communication cost $C1$ versus the split training communication cost. For simplicity in examining communication overhead trends, I still assume that the federated aggregation communication cost is related to the number of neurons. In addition, I set χ as a weighting parameter indicating the split training communication overhead versus the number of neurons for different amounts of training data. Thus, $C2 \triangleq \chi \times number\ of\ neurons$. The increase of χ implies an increase in the amount of training data.

It can be seen in Figure 3.6, similar to Figure 3.5, that the FL communication cost is independent of the amount of training data and thus remains fixed to a constant value. As χ increases, the communication cost of proposed MSFTL and

MSFTL without TL also increases. Moreover, in case χ is small, our proposed MSFTL achieves less communication cost, otherwise, FL achieves less. This is because as the amount of training data increases, the number of samples transmitted by the vehicle to the edge for training increases. Therefore, the communication cost incurred during forward propagation versus backpropagation communication is increasing. Furthermore, our proposed MSFTL always has less communication cost than without TL due to the reduced times of backpropagation.

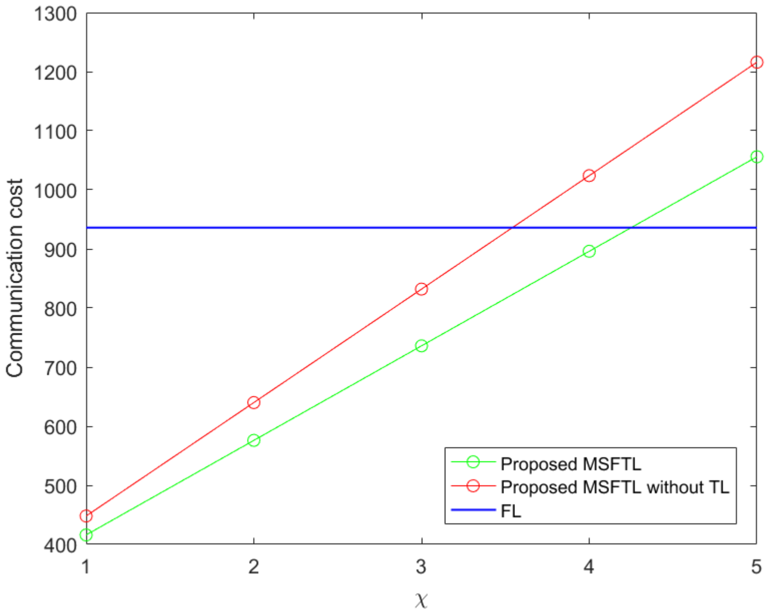


Figure 3.6: Communication cost of different frameworks.

Figure 3.7 evaluates the performance of the novel TL approach for the proposed learning framework in the presence of sparse training samples. The proposed MSFTL is comparable to the MSFTL without TL in the case of only one vehicle. It can be viewed from the figure that as the number of samples increases, all the frameworks' accuracy increases. However, compared to the MSFTL without TL, the MSFTL achieves a performance that far exceeds MSFTL without TL accuracy. This demonstrates the significant contribution of the proposed TL-based learning approach to improving the system performance in the case of sparse training samples.

3.5.1 Optimisation mechanism

I show the simulation results in evaluating the performance of our optimisation mechanism in this subsection. To demonstrate the effectiveness of our game theoretical mechanism more intuitively, I assume all vehicles involved in the training have the same conditions (such as CPU cycles, velocity etc. Thus, in case Eq. (3.32) equals 0, Eq. (3.33) can be written as

$$U_m = \gamma ln(1 + \theta \frac{\alpha R_m (I-1)}{I_m H_{n,m}}) - \delta R_m. \tag{3.5}$$

I set $\gamma = 0.13$, $\alpha = 10$, $\delta = 0.08$ and $\theta = 8.5$ to approximate the simulation results in Figure 3.4. The maximum accuracy is set as 98% and the training epoch is set as 100 simulation results above. Similarly, the data set is divided into 100 parts, $D_{n,m}^{min} = 1$ and $D_{n,m}^{max} = 2.5$. In addition, I use $H_{n,m}$ to denote the data unit training cost and $\Gamma_{(n,m)} = 20$ s. The computation capability f_m allocated to each vehicle is 3 Gcycles/s [39] and computational size required $d_{P_{2,3}^m} + d_{P_4^m}$ of TEC m is 30 MB.

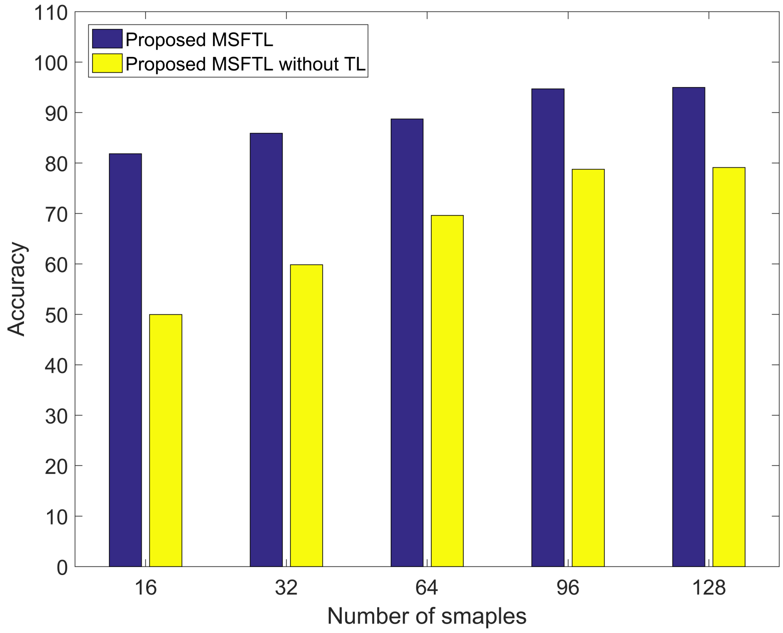


Figure 3.7: Accuracy of different frameworks with sparse samples.

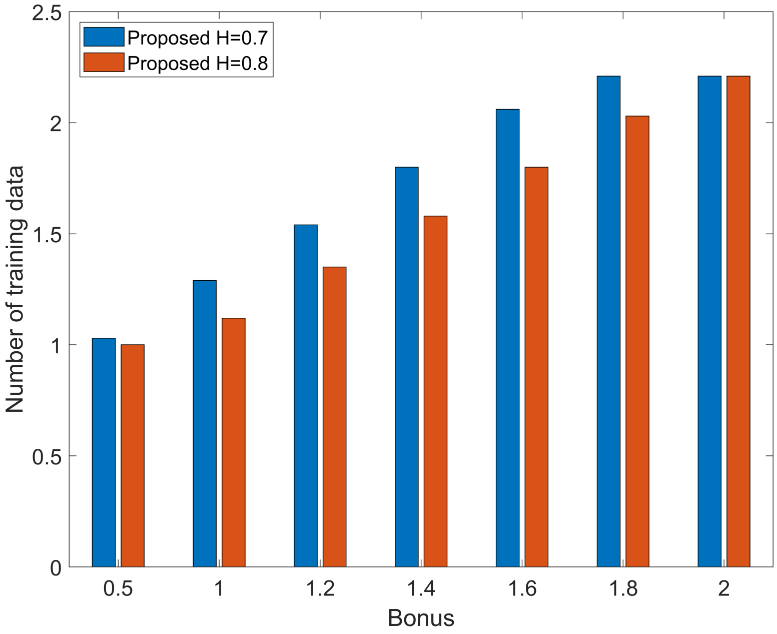


Figure 3.8: Reward impact on training data number.

In Figure 3.8, I investigate the influence of bonuses on the number of training data in different unit costs. I assume that the residence time of all vehicles is sufficient. It is seen that vehicles are less to participate in training at low bonus values. Because low bonus results in low motivation. As the bonus value increases, the vehicles perform more training data, with higher-cost vehicles willing to train fewer data. Eventually, the same amount of data is trained and remains the same for vehicles with different unit costs. This is because, at a high bonus value, the TEC is limited by the maximum accuracy, so the amount of training data no longer changes.

Figure 3.9 illustrates the variation in training unit cost for different residence times and mechanisms. It is seen that the vehicle does not have enough time to train the most appropriate amount of data at a short residence time and therefore vehicles with different costs provide the same training data. The amount of data increases as the residence time increases, but the proposed mechanism in different costs reaches stability successively at different residence times. This is because the optimal

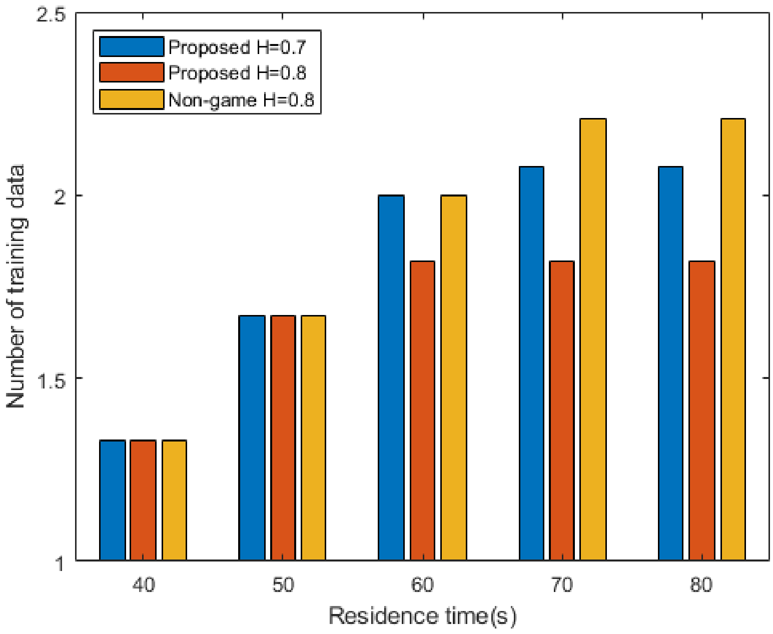


Figure 3.9: Total training time versus various residence time.

number of data for vehicle participation in training has been reached. The method without the game continues to grow and results in more energy costs. Moreover, our mechanism is less than or equal to the non-game theoretical mechanism in all cases. This demonstrates the effectiveness of our mechanism in reducing energy costs.

3.6 Summaries

In this chapter, I designed a new vehicle SemCom framework, named MSFTL. It divides the trained DLJSC coder into four parts and utilises the proposed split federated learning for training, which can adapt to complex and various vehicle offloading scenarios. Further, in the proposed framework, I presented a novel approach based on TL to speed up training as well as increase its accuracy. In particular, this approach performs excellently in a low training sample environment and reduces computing and communication costs. Moreover, an efficient high-mobility energy optimisation mechanism for MSFTL was proposed. It was designed based on the Stackelberg game theoretic by jointly taking into account vehicle mobility and semantic model accuracy. I have also conducted simulation experiments to evaluate our proposed framework and energy optimisation mechanism. The simulation results demonstrated the effectiveness of our learning framework and mechanism. In the next chapter, the optimisation mechanisms for one of the extended 6G networks, i.e., air-terrestrial networks, will be investigated.

Chapter 4

Air-terrestrial SemCom System

4.1 Introduction

In this section, the challenges of SemCom in air-terrestrial networks are analysed and investigated. We first summarised several significant outstanding challenges for SemCom in AENs. First, the implementation of SemCom in AENs raises the sophisticated network energy optimisation challenge. Because SemCom shifts part of the communication load to the computational load to increase spectral efficiency. The transformation of energy utilisation locations poses an extra energy optimisation issue to AENs that inherently require energy efficiency improvements. How to develop an energy-efficient SemCom architecture for the air network and how to optimise the energy efficiency of SemCom is hence an essential concern.

Furthermore, SemCom requires real-time updating ML-based semantic coders for various specific content [7]. The existing FL framework for updating semantic coders in general networks [11], [12] however faces several challenges in AENs. For instance, the distributions of training data from different coder owners are frequently not independent and identically distributed (Non-IID) [56]. Furthermore, as the AEN is sophisticated and AECs are energy-limited, the energy efficiency of the learning framework has to be considered. How to timely update the semantic coder accurately and energy-efficiently in an AEN with Non-IID training data thus is a challenge for SemCom to apply in AENs.

In this chapter, we propose a novel energy-efficient SemCom system for AENs. We also discuss the resource allocation problem during SemCom usage. A new EGTIM based on the proposed system is presented to optimise the network energy efficiency fairly. In addition, we propose a GEDL framework for semantic coders updating in AENs. It renews the proposed EGTIM and combines EGTIM with a conventional distributed learning approach to update semantic coders accurately and efficiently in terms of energy consumption.

The remainder of this chapter is organised as follows. We describe the proposed system model in Section 4.2. In Section 4.3, the game problem formulation and the proposed EGTIM are presented. Section 4.4 describes the presented GEDL framework for semantic coder updating in AENs. Simulation results are shown in Section 4.5. Finally, we conclude this chapter in Section 4.6.

4.2 System model

In this chapter, I consider a three-dimensional edge network aided by an AEC j (Figure 4.1). The TECs provide edge services via semantic coders to subscribers on the terrestrial. An AEC j with semantic coders hovers in the air and assists TECs to provide edge services to subscribers. It communicates with subscribers via TECs which act as relay nodes. The network thus does not share the same spectrum resources between AEC-TECs and TECs-subscribers. Furthermore, to optimise the allocation of network energy resources, semantic extraction task locations allow for replacement by conventional communication transmission, followed by SemCom calculations and transmission to the subscribers.

I assume that the energy power of AEC j lingers in the air is P_j^l. The free computational capability (free CPU-cycle frequency) of AEC j is f_j. Moreover, there are I TECs within the service range of AEC j that provide edge service to subscribers. I denote the data size of tasks that each TEC i prepare to transmit to

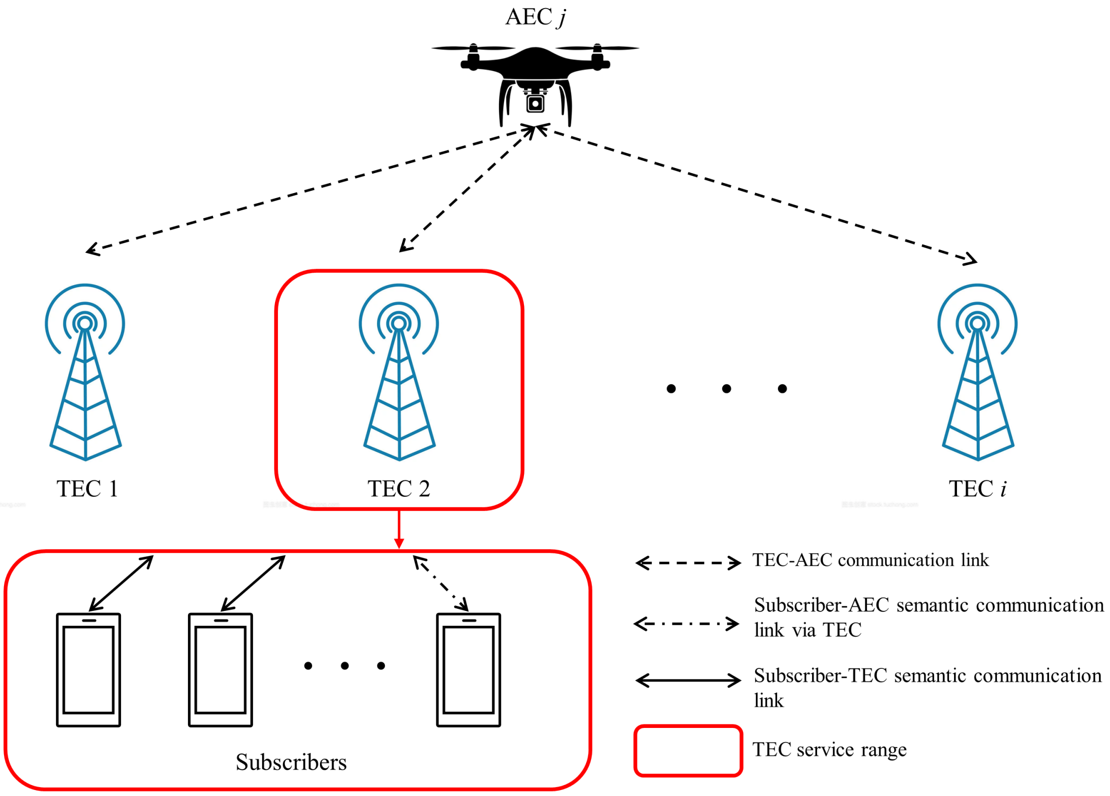

AEC *j*
TEC 1
TEC 2
TEC *i*
Subscribers
TEC-AEC communication link
Subscriber-AEC semantic communication link via TEC
Subscriber-TEC semantic communication link
TEC service range

Figure 4.1: Proposed system model.

subscribers as m_i bits. The semantic encoder execution latency of TEC i for these tasks can be expressed as:

$$T_i^C = \frac{am_i}{f_i}. \tag{4.1}$$

where f_i is the CPU-cycle frequency of TEC i to process these semantic compression tasks and the unit is cycles/s. Further, a is the pure number of CPU-cycle consumed to calculate each 1-bit [57]. According to [38], the computing power of the TEC i can be denoted by

$$P_i^C = \kappa f_i^3, \tag{4.2}$$

where κ is the CPU architecture-related coefficient. I thus have the execution energy consumption of TEC i for these semantic compression tasks as:

$$E_i^C = \kappa a m_i f_i^2. \tag{4.3}$$

Similarly, in the case of the TEC i provides part of the semantic compression task bits $m_{i,j}$ to the AEC j, the execution latency and energy consumption of AEC j can be expressed as:

$$T_j^C = \frac{am_{i,j}}{f_{j,i}}, \tag{4.4}$$

$$E_j^C = \kappa am_{i,j} f_{j,i}^2, \tag{4.5}$$

where $f_{j,i}$ is the CPU-cycle frequency that AEC j allocate to the task bits $m_{i,j}$. To ensure the QoS, in this chapter, we assume $f_{j,i} = f_i$. In addition, during the semantic compression task providing process, the data transmission rate of the TEC i to the AEC j can be denoted by

$$r_i^T = B_i \log_2(1 + \frac{p_i g_i}{\sigma^2}), \tag{4.6}$$

where B_i is the bandwidth of the communication channel between the TEC i and the AEC j. Further, p_i, g_i and σ are the transmission power, channel gain and additive white Gaussian noise (AWGN) power in this channel, respectively. I then can have the transmission delay as:

$$T_i^T = \frac{m_{i,j}}{r_i^T} = \frac{m_{i,j}}{B_i \log_2(1 + \frac{p_i g_i}{\sigma^2})}. \tag{4.7}$$

Thus, the transmission energy consumption is

$$E_i^T = p_i T_i^T = \frac{p_i m_{i,j}}{B_i \log_2(1 + \frac{p_i g_i}{\sigma^2})}. \tag{4.8}$$

As the completed semantic extraction task result size is much smaller than the task size. Resembling [58], [39], we hence ignore the transmit delay and energy consumption of transmission tasks after semantic compression.

4.3 Stackelberg game theoretic mechanism design

To improve the AEN energy efficiency, the fairness optimising assignment of the number of semantic compression tasks processed by the TECs and the AEC is essential. I identify that when AEC edge resources are underutilised, more energy is consumed on air hover. This results in a significant amount of energy being wasted rather than performing edge services. Therefore, I construct the TECs and the AEC interaction as a Stackelberg game [52] from the economic perspective. It incentivises TECs to provide partial semantic extraction tasks to the AEC in fairness, where the AEC is trusted, thus improving the network energy efficiency. The Stackelberg game is comprised of a leader and followers, where the followers change their policies according to the policies developed by the leader. Thus, the proposed incentive mechanism consists of the game at the AEC (leader) and the game at TECs (followers), which I elaborate on in detail in the following two subsections.

4.3.1 Game at the AEC

Without loss of generality, I define the monetary utility U_j of the AEC j as:

$$U_j = N_j + R_j - B_j - G_j. \tag{4.9}$$

where N_j is the net income of AEC j to transmit semantic compression tasks to subscribers and R_j is the additional energy cost revenue of AEC j gained as a result of performing provided semantic compression tasks from TEC i. Further, G_j is the gain loss of AEC j due to the transfer of some holdup energy to the additional semantic transmission execution resulting in a reduction of the holdup time. Moreover, B_j is the bonus paid to TECs providing the tasks. I consider the monetary salary N_j as the energy consumption similar to the previous study [59]. Thus, I have

$$N_j(\mathbf{m_{i,j}}) + R_j(\mathbf{m_{i,j}}) = (\alpha + \beta) \sum_{i=1}^{I} \kappa a m_{i,j} f_{j,i}^2, \tag{4.10}$$

where α is the net income monetary parameter and β is the energy cost monetary parameter.

The gain loss G_j depends on the aerial hover time and I define it as gain loss of not performing its regular tasks. To obtain the G_j, I first formula the residence time of AEC j without additional semantic compression tasks as:

$$T_j^0(\mathbf{m_{i,j}}) = \frac{E_j}{P_j^l + P_j^n + \kappa f_{j0}^3}, \tag{4.11}$$

where E_j is the hover energy of AEC j and f_{j0} is the CPU-cycle frequency required for the AEC j to perform its regular tasks. Further, P_j^n is the AEC utilising power with no economic benefit. I then have the residence time of AEC j with additional semantic compression tasks as:

$$T_j^1(\mathbf{m_{i,j}}) = \frac{E_j - e_j}{P_j^l + P_j^n + \kappa f_{j0}^3}, \tag{4.12}$$

where $e_j = \sum_{i=1}^{I} \kappa a m_{i,j} f_{j,i}^2$ is the energy consumption of the AEC j to execute the provided tasks. Therefore, I can find the G_j as:

$$G_j(\mathbf{m_{i,j}}) = \gamma \kappa f_{j0}^3 (T_j^0 - T_j^1), \tag{4.13}$$

where γ is the income monetary parameter. As the energy benefit that would have been gained by the sale disappears, $\gamma = \alpha + \beta$.

In addition, I set the unit price of each task bit being transmitted from the TEC to the AEC to b. The bonus paid B_j to TECs providing the tasks can be expressed by

$$B_j(b, \mathbf{m_{i,j}}) = \sum_{i=1}^{I} b m_{i,j}. \tag{4.14}$$

Therefore, I have

$$U_j(b, \mathbf{m_{i,j}}) = (\alpha + \beta) \sum_{i=1}^{I} \kappa a m_{i,j} f_{j,i}^2 - \sum_{i=1}^{I} b m_{i,j} - \gamma \kappa f_{j0}^3 (T_j^0 - T_j^1). \tag{4.15}$$

Mathematically, the AEC's game problem can be presented as:

Problem 4.1:

$$\max_{b} \quad (\alpha + \beta) \sum_{i=1}^{I} \kappa a m_{i,j} f_{j,i}^2 - \sum_{i=1}^{I} b m_{i,j} - \gamma \kappa f_{j0}^3 (T_j^0 - T_j^1) \tag{4.16a}$$

$$s.t. \quad \sum_{i=1}^{I} f_{j,i} \leq f_j \tag{4.16b}$$

$$b > 0 \tag{4.16c}$$

$$E_j > e_j. \tag{4.16d}$$

4.3.2 Game at TECs

Similarly, I can define the utility of a TEC i as:

$$U_i = B_j + C_i^c - N_i - C_i^t - S_i. \tag{4.1}$$

I will explain the meaning of this formula in turn. First, B_i is the bonus gain of TEC i from the AEC j. Based on Eq. (4.14), I have

$$B_j(\mathbf{b}, m_{i,j}) = b m_{i,j}. \tag{4.2}$$

Further, C_i^c is the revenue of the saved computing energy cost of TEC i. As it not performing the provided task locally and save the cost. I can express C_i^c by

$$C_i^c(m_{i,j}) = \beta \kappa a m_{i,j} f_i^2. \tag{4.3}$$

The N_i from Eq. (4.17) is the net income forgone of TEC i to transmit semantic compression tasks to subscribers. The net income is transferred to the AEC. Therefore, similar to Eq. (4.10), I have the net income forgone of TEC i as:

$$N_i(m_{i,j}) = \alpha\kappa a m_{i,j} f_i^2. \tag{4.4}$$

In addition, C_i^t is the transmission energy income loss from the TEC i to the AEC. As no economic benefit is generated from this energy, I denoted the C_i^t by

$$C_i^t(m_{i,j}) = \gamma \frac{p_i m_{i,j}}{B_i \log_2(1 + \frac{p_i g_i}{\sigma^2})}. \tag{4.5}$$

Particularly, S_i is set as the satisfaction revenue change of TEC i due to the semantic transmission tasks transfer from the TEC to the AEC. The lower satisfaction results in a lower motivation for subscribers to access the edge services, resulting in lower gains. In this chapter, I argue that subscriber satisfaction is related to task processing delay. I hence model the satisfaction revenue as a logarithmic function related to execution delay. Because the logarithmic function based on execution delay precisely expresses the satisfaction of subscribers with the edge services [60], [61]. The S_i can be denoted by

$$S_i(m_{i,j}) = \varphi(\ln(1 + \theta - T_i^C) - \ln(1 + \theta - T_i^C - T_i^T)), \tag{4.6}$$

where φ is the monetary parameter and $\theta \leq T_i^C + T_i^T$ to ensure the satisfaction is positive. Therefore, I have

$$\begin{aligned} U_i(\mathbf{b}, m_{i,j}) = b m_{i,j} + (\beta - \alpha)\kappa a m_{i,j} f_i^2 - \gamma \frac{p_i m_{i,j}}{B_i \log_2(1 + \frac{p_i g_i}{\sigma^2})} \\ - \varphi(\ln(1 + \theta - T_i^C) - \ln(1 + \theta - T_i^C - T_i^T)). \end{aligned} \tag{4.7}$$

Problem 4.2:

$$\max_{m_{i,j}} \quad b m_{i,j} + (\beta - \alpha)\kappa a m_{i,j} f_i^2 - \gamma \frac{p_i m_{i,j}}{B_i \log_2(1 + \frac{p_i g_i}{\sigma^2})}$$

$$- \varphi(\ln(1 + \theta - T_i^C) - \ln(1 + \theta - T_i^C - T_i^T)) \tag{4.8a}$$

$$s.t. \quad 0 \leq m_{i,j} \tag{4.8b}$$

$$p_i^r \leq \zeta \tag{4.8c}$$

Algorithm 4.1 EGTIM

```
Initialization: semantic transmission tasks $m_i$, CPU-cycle frequency $f_i$, the maximum number of iteration $K$, the stopping criterion threshold $\xi > 0$, and learning rate $\theta$
for each $i = 1, 2, ..., I$
    Derive optimal $m^*_{i,j}$, i.e., $f_i(b)$ by $\frac{\partial U_i}{\partial m_{i,j}} = 0$
end for
Substitute $f_i(b)$ in $U_j(b)$
while $k < K$
    $b^{'} = b - \theta \bigtriangledown U_j(b)$
    $b^{''} = b, b = b^{'}$
    until $b^{''} - b < \xi$
end while
Derive optimal $m_{i,j}$ according to optimal $b$
return $b$ and $m_{i,j}$
```

where p^r_i is the TEC i's privacy concern and ζ is the privacy leakage threshold. Because even though the AEC is trusted, setting a TEC privacy breach tolerance value is necessary to prevent possible attacks. It indicates the maximum acceptable providing task bits. According to [62], I can denote the relationship between transfer tasks bits and privacy leakage value as:

$$p^r_i = \log_2(1 + e^{1-\frac{m_i+1}{m_{i,j}}}). \tag{4.1}$$

4.3.3 Nash equilibrium for the game

The game of TECs and the AEC can model as a Stackelberg game. To guarantee fairness, the objective of the TECs is to maximise their utility by simultaneously selecting the most appropriate $m_{i,j}$ when given the known unit price b. Meanwhile, the AEC's objective is to maximise its utility by varying b, for a known $m_{i,j}$. The game can be expressed by

$$U_i(\mathbf{b}^*, m_{i,j}^*) \geq U_i(\mathbf{b}^*, m_{i,j}), \tag{4.2}$$

$$U_j(b^*, \mathbf{m_{i,j}^*}) \geq U_j(b, \mathbf{m_{i,j}^*}), \tag{4.3}$$

Where b^* and $m_{i,j}^*$ are solutions in which the parties jointly pursue the optimal strategies, i.e., the NE point(s). First, I demonstrate the existence of NE in this game.

Existence of NE:

The second-order partial derivative of $U_i(\mathbf{b}^*, m_{i,j})$ can be denoted by

$$\frac{\partial^2 U_i}{\partial m_{i,j}^2} = \varphi((\frac{\frac{a}{f_i}}{\theta - T_i^C + 1})^2 - (\frac{\frac{a}{f_i} + \frac{1}{r_i^T}}{\theta - T_i^C - T_i^T + 1})^2). \tag{4.4}$$

Since $\theta - T_i^C + 1 > \theta - T_i^C - T_i^T + 1$ and $\frac{a}{f_i} < \frac{a}{f_i} + \frac{1}{r_i^T}$. I can observe that $\frac{\partial^2 U_i}{\partial m_{i,j}^2} < 0$. Hence, U_i is concave in $m_{i,j}$. As the strategy set of the TEC i is also compact and convex, based on the Debreu-Glicksberg-Fan theorem [52], the NE of this game exists.

In order to achieve NE, I utilise the backward induction approach in game theory and obtain the optimal strategies of followers (TECs) first. Subsequently, based on these TECs' strategies, the leader's (AEC's) optimal strategy is developed. Thus, I first derive the first-order partial derivative of U_i as:

$$\begin{aligned} \frac{\partial U_i}{\partial m_{i,j}} &= b + (\beta - \alpha)\kappa a f_i^2 - \gamma \frac{p_i}{r_i^T} \\ &- \frac{\varphi f_i^2(\theta + 1)}{(f_i - a m_{i,j} + \theta f_i)(r_i^T f_i - f_i m_{i,j} - r_i^T a m_{i,j} + r_i^T \theta f_i)}. \end{aligned} \tag{4.5}$$

As U_i is concave in $m_{i,j}$, the maximum of U_i and corresponding $m_{i,j}$ thus can be derived by $\frac{\partial U_i}{\partial m_{i,j}} = 0$. Due to it being hard to be expressed, I simply denoted the optimal $m_{i,j}^* = f_i(b)$. Therefore, the utility function of U_j can be rewritten as:

$$U_j(b) = (\alpha + \beta) \sum_{i=1}^{I} \kappa a f_i(b) f_{j,i}^2 - \sum_{i=1}^{I} b f_i(b) - \gamma \kappa f_{j0}^3 (T_j^0 - T_j^1). \tag{4.6}$$

If I can derive the maximum U_j and corresponding b, I therefore can obtain the corresponding $m_{i,j}^*$ in a closed-form based on Eq. (4.29). However, due to the complexity of Eq. (4.30), I cannot derive the NE closed form. Fortunately, b and $m_{i,j}$ both have boundaries. The NE thus can be obtained by performing a gradient descent method [63] over b and $m_{i,j}$. The solution step is shown in Algorithm 4.1.

4.4 Efficient distributed Learning Design

The application of SemCom significantly improves the network QoS. Nevertheless, how to update users' ML-based semantic coders efficiently and accurately in real-time becomes one of the biggest challenges of SemCom studies. FL is a potential approach to cope with the challenge of semantic coder updates in the network [12]. Nevertheless, the 3D network environment is sophisticated, and energy limited. In particular, the case where the users' training data are Non-IID significantly reduces the SemCom QoS. FL thus is not the optimal solution for AENs.

To address these challenges, I propose a GEDL framework for AENs (Figure 4.2). Specifically, TECs first transmit some Non-IID SemCom transmission tasks to the AEC based on our proposed renewed EGTIM for semantic coder updating. The TECs then update the semantic coder based on their training data and transmit the new coder model to the AEC for the federated aggregation. Subsequently, the AEC performs the federated aggregation and retrains the aggregated model utilising the tasks provided by TECs. This is because AEC is flexible in terms of data collection, it is often used as a federated aggregation node [64]. Finally, the AEC sends back the model to participate in TECs and complete one training epoch. The model accuracy thus can be improved while maximising energy efficiency. I will demonstrate these in our simulations.

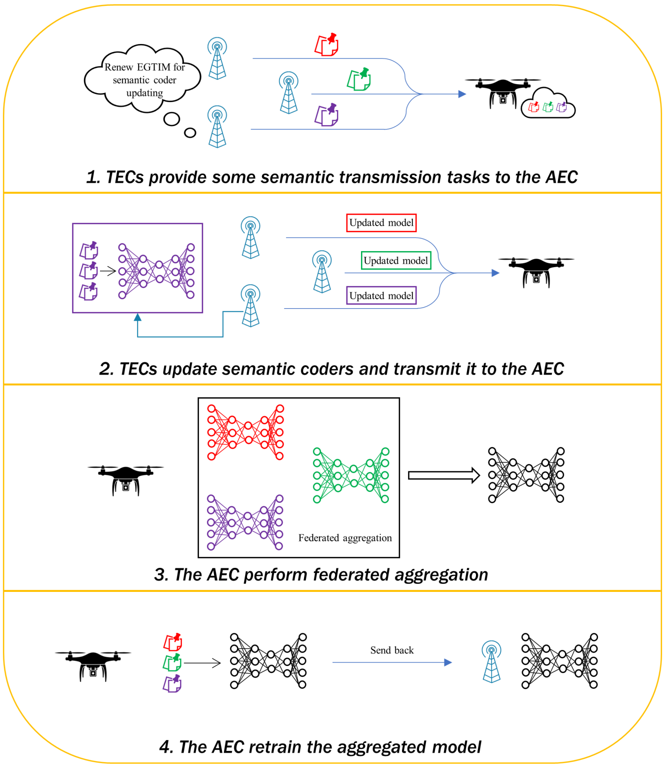

Renew EGTIM for semantic coder updating
1. TECs provide some semantic transmission tasks to the AEC
Updated model
Updated model
Updated model
2. TECs update semantic coders and transmit it to the AEC
Federated aggregation
3. The AEC perform federated aggregation
Send back
4. The AEC retrain the aggregated model

Figure 4.2: The process of proposed GEDL.

I first renew the EGTIM for semantic coder updating. As increased semantic coder accuracy can improve the network QoS, it enhances network revenue. Similar to [65], I utilise a logarithmic function to model the relationship between training accuracy and training task size. The revenue of model accuracy improvement thus can be denoted by

$$A_j^t = \delta(\ln(1 + \sum_{i=1}^{I} m_{i,j}^t) + \eta), \tag{4.7}$$

where $m_{i,j}^t$ is the proving task bits from the TEC i to the AEC j before training and δ is the monetary parameter. Further, η is the basic accuracy of FL.

Therefore, I should update the utility function of the AEC j as:

$$U_j^t = A_j^t + R_j^t - B_j^t - G_j^t. \tag{4.8}$$

Similar to Eq. (4.9), in Eq. (4.32), R_j^t is the additional energy cost revenue of

AEC j gained during training and B_j^t is the bonus paid from the AEC j to TECs providing the tasks. Further, G_j is the gain loss of the AEC j due to the transfer of some holdup energy to additional training.

Therefore, the game problem for AEC j when coder training can be presented as:

Problem 4.3:

$$\begin{aligned}
\max_{b} \quad & \delta(\ln(1+\sum_{i=1}^{I} m_{i,j}^t)+\eta)+\beta\kappa a f_j^{t^2}\sum_{i=1}^{I} m_{i,j}^t \\
& -\sum_{i=1}^{I} b m_{i,j}^t-\gamma\kappa f_{j0}^3(T_j^0-T_j^1) && (4.9a)\\
s.t. \quad & f_j^t \le f_j && (4.9b)\\
& b > 0 && (4.9c)\\
& E_j \ge e_j && (4.9d)
\end{aligned}$$

where f_j^t is the CPU-cycle frequency of the AEC j to perform the additional training after federated aggregation. Due to the requirement to perform federated aggregation, the power of AEC j for the regular task without economic benefit also needs to be plus the aggregation power. Furthermore, the reduction in training sample size reduces the model accuracy and thus affects the accuracy of the model after federated aggregation. Therefore, TECs still train the number of new tasks they have. The utility function of proving semantic transmission tasks thus can be changed from Eq. (4.17) by

$$U_i^t = B_i^t - C_i^t ra - S_i^t. \tag{4.1}$$

where B_i^t is the training bonus gain of TEC i from the AEC j and C_i^{tra} is the transmission energy consumption. Further, S_i^t is the revenue change due to the satisfaction change. As satisfaction is associated with training time, I have

$$S_i^t = \varphi(\ln(1+\theta^t - T_i^t) - \ln(1+\theta^t - T_i^t - T_i^a)), \tag{4.2}$$

where T_i^t is the distributed learning training computing time without AEC additional training, i.e., FL training computing time. Further, T_i^a is the AEC additional training time. Since the training time tends to be much greater than the training data transmission time, I ignore the variation in satisfaction due to the transmission time. Hence, I have the game problem for the TEC i during training new coders as:

Problem 4.4:

$$\begin{aligned}
&\max_{m_{i,j}^t} \quad bm_{i,j}^t - \gamma \frac{p_i m_{i,j}^t}{B_i \log_2(1+\frac{p_i g_i}{\sigma^2})} - \varphi(\ln(1+\theta^t - T_i^t) \\
&- \ln(1+\theta^t - T_i^t - T_i^a)), && (4.3a)\\
&s.t. \quad 0 \leq m_{i,j} && (4.3b)\\
&\qquad \varrho \log_2(1+e^{1-\frac{1+m_{i,j}^t}{m_{i,j}^t}}) \leq \zeta && (4.3c)
\end{aligned}$$

where ϱ is the weight parameter. It measures the increased risk of privacy leakage arising from the transmission of $m_{i,j}^t$ as it relates to the new coder. Furthermore, m_i^t is the total training task bits of the TEC i. It can be found from Problem 4.4 that the strategy set of the TEC i is also compact and convex as same as Problem 4.2. In addition, the second differentiation of U_i^t is similar to U_i and concave in $m_{i,j}^t$. Thus, the NE of this game is still existing and the NE point can be achieved by Algorithm 4.1.

4.5 Simulation results

In this section, I provide simulation results to validate the performance of the proposed EGTIM and GEDL. First, I elaborate on the energy efficiency of our EGTIM. The advantage of our GED framework is then assessed by comparing it with baseline distributed learning in image transmission scenarios [11], [12].

4.5.1 EGTIM

I first elaborate on the simulation settings in assessing the performance of our proposed EGTIM. I assume there are 5 TECs in the service range of the AEC j. To better demonstrate our proposed mechanism, I assume that all TECs have the same conditions. Similar to [38] and [39], I set $a = 120$; $p_i = 0.2w$; $\kappa = 10^{-26}$; $f_i = 0.5 \times 10^9 cycles/s$; $f_{j0} = 0.5 \times 10^9 cycles/s$. Further, I assume the monetary parameter $\alpha = 1$, $\beta = 1$ and thus $\gamma = 2$. If not mentioned, the hold-up power of the AEC is set as 1 w and by default the constraints are all satisfied.

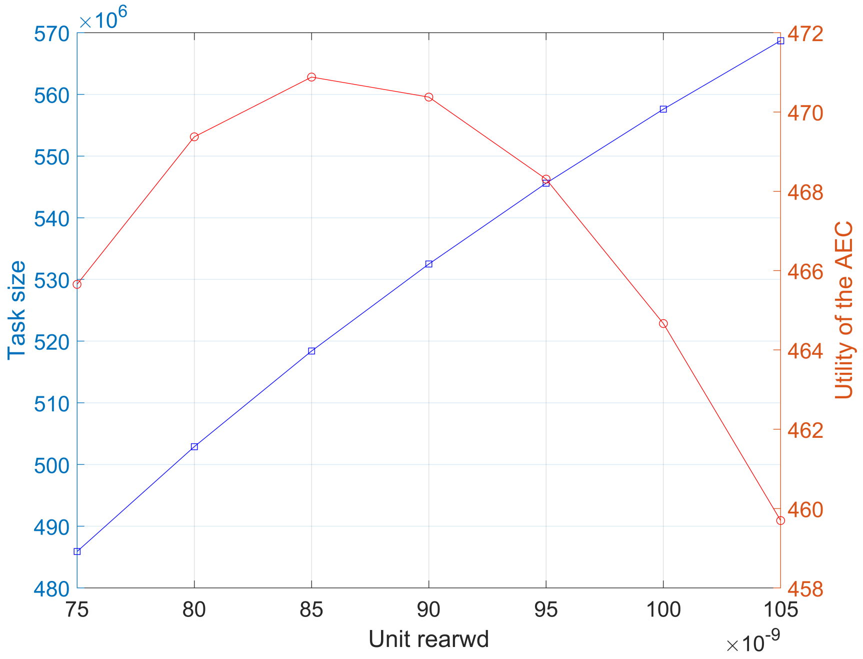


Figure 4.3: NE existence under the proposed EGTIM.

In Figure 4.3, the existence of NE is demonstrated. It can be observed that as the unit reward value increases, the optimal task size that TECs are willing to provide also increases. This is due to the increased transfer task size allowing TECs to earn greater benefits as the unit rewards increase. However, the utility function of the AEC shows an increasing trend followed by a decreasing trend. There is therefore an NE point that maximises the utility of the ATC while ensuring that the utilities of TECs are maximised (i.e., optimal transfer task size).

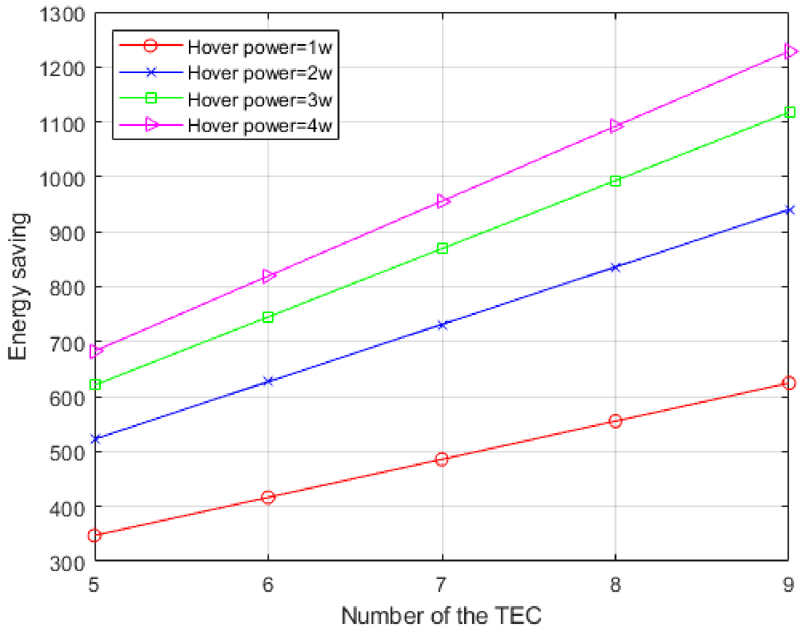


Figure 4.4: Energy saving of proposed EGTIM in various scenarios.

Figure 4.4 illustrates the energy savings in joules (J) at different amounts of TECs and different hover consumption power. I define energy saving as the reduction in wasted hover consumption minus the lost energy consumption for regular AEC tasks and the power consumption of TECs transmitting. Mathematically, the energy saving equals $\frac{R_j}{\beta} - \frac{G_j + C_i^t}{\gamma}$. As can be observed, more energy can be saved as the number of TECs increases. This is due to the fact that the increase in the number of TECs decreases the energy consumption in hover and outweighs the resulting loss raise. It is notable that the number of TECs does not grow indefinitely as the AEC has a finite computing capacity. In addition, the higher the hover power, the greater the energy saving, but the magnitude of the increase is decreasing. Because the hover power increase means consuming the same energy for additional semantic transmission tasks, the AEC can be maintained on air for a longer time. The corresponding cost loss thus falls and the magnitude of the increase is decreasing as the percentage of hover energy consumption of the AEC becomes larger.

In Figure 4.5, I evaluate the influence of different CPU-cycle on providing task

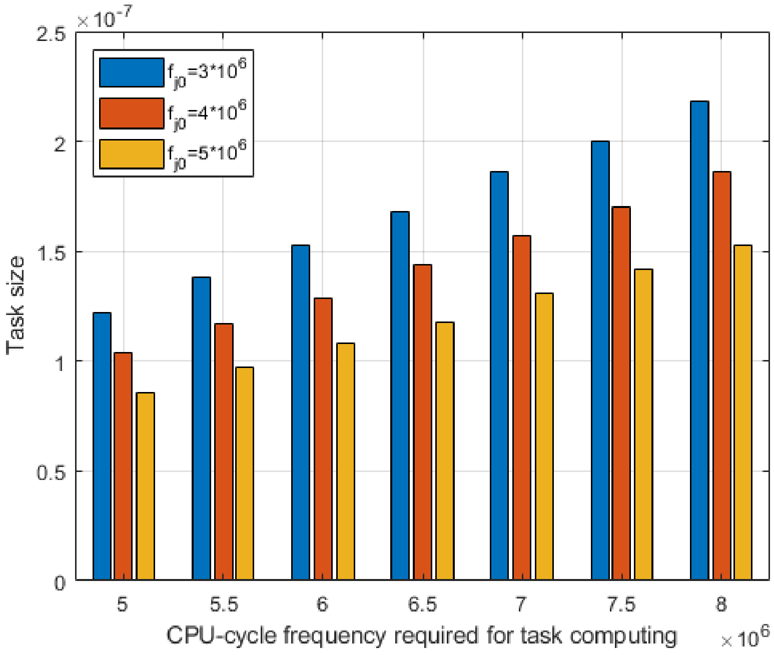


Figure 4.5: Effect of different CPU-cycle on providing task size.

size from TECs to the AEC. It is observed that more CPU-cycle frequency required for semantic task transmission makes TECs more inclined to transfer more task bits. However, the increase in CPU-cycle frequency required for regular tasks results in lower providing task sizes. This is because the increased CPU-cycle frequency required for tasks increases the efficiency of AEC hover energy utilisation. Therefore, TECs are biased towards providing more tasks for more revenue. Further, the increased f_{j0} increases the hover time reduction benefit loss and therefore reduces the overall data transfer revenue and hence the unit reward.

4.5.2 GEDL

To estimate our GEDL, I employ the convolutional neural network (CNN) as the semantic coder and set the application scenario as an image transmission environment, similar to [17]. Further, I train models on the CIFAR-10 [54] dataset with 60000 training data and 10000 test data, which all have 10 class images. As in the same previous subsection, I assume there are 5 TECs involved in the training.

To create the Non-IID training environment, I enable each TEC in training to have only four classes of the training data in the different 10000 CIFAR-10 data. The transmission accuracy is determined by the PSNR, which is a criterion for the quality of image transmission in SemCom [17]. I have

$$PSNR = 10lg\frac{MAX^2}{\|x - \hat{x}_j\|^2}, \tag{4.1}$$

where MAX is the maximum value for a pixel and x is the input of the image and $\hat{x}_j$ is the output via the semantic coder.

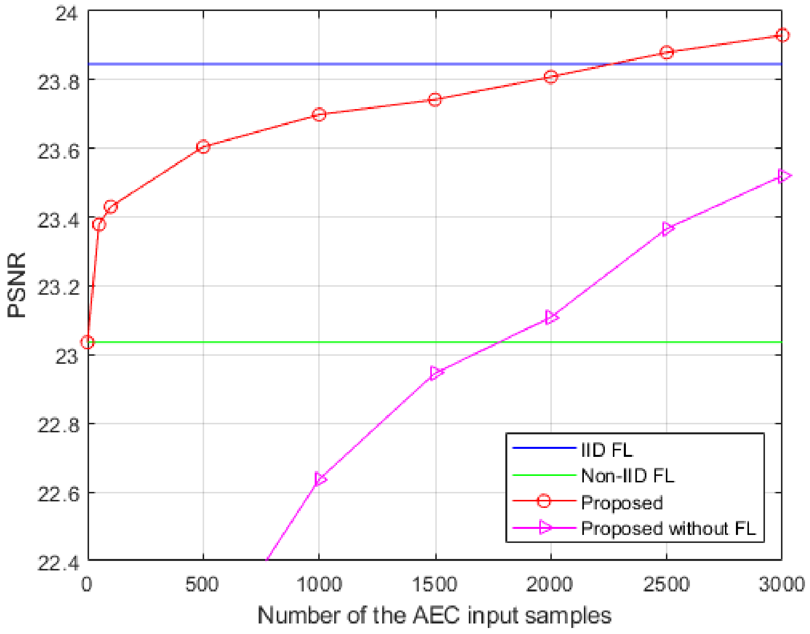


Figure 4.6: The accuracy of various training frameworks with the AEC input samples grows.

Figure 4.6 demonstrates the comparison of accuracy under different learning frameworks. I compare the different learning frameworks together when the training data is IID. Furthermore, I also add the FL model with IID training data as a reference. It is seen that as the training data obtained by the AEC increases, the coder accuracy also increases. In particular, the trend of the increase exhibits a trend of the logarithmic function, thus verifying our hypothesis in Eq. (31). In addition, with the increase in the volume of data, the accuracy of the proposed GEDL increased and even exceeded the performance of FL trained with the IID model. The accuracy of our proposed GEDL without FL also rapid growth. This is because the greater the amount of data AEC has, the more the training process approaches central learning. The training data is mixed together for training and therefore the accuracy increases. Nevertheless, it is noteworthy that due to privacy, AEC's available computing resources and energy constraints, the data AEC obtains is limited. However, our proposed GEDL is always more accurate than FL with the Non-IID training scenario.

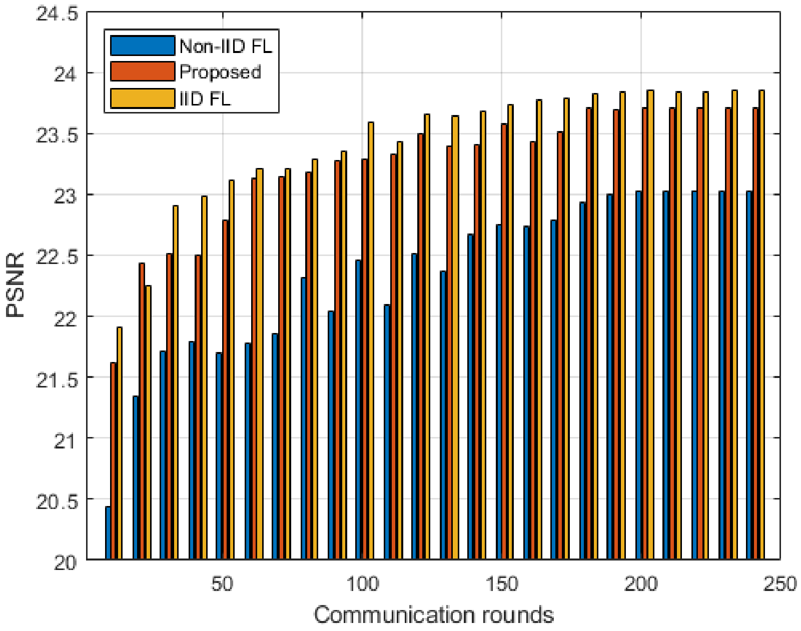


Figure 4.7: Convergence speed of different training frameworks.

Figure 4.7 shows the comparison of the convergence speed of FL and our proposed distributed learning. I also included FL trained with the IID data as a reference. It can be observed that all learning eventually reaches convergence and the time to reach convergence is almost the same. However, our proposed GEDL is always more accurate than FL after each communication round. This is because our proposed GEDL is based on the FL for accuracy improvement and thus it increases the training accuracy but needs the FL process to reach convergence.

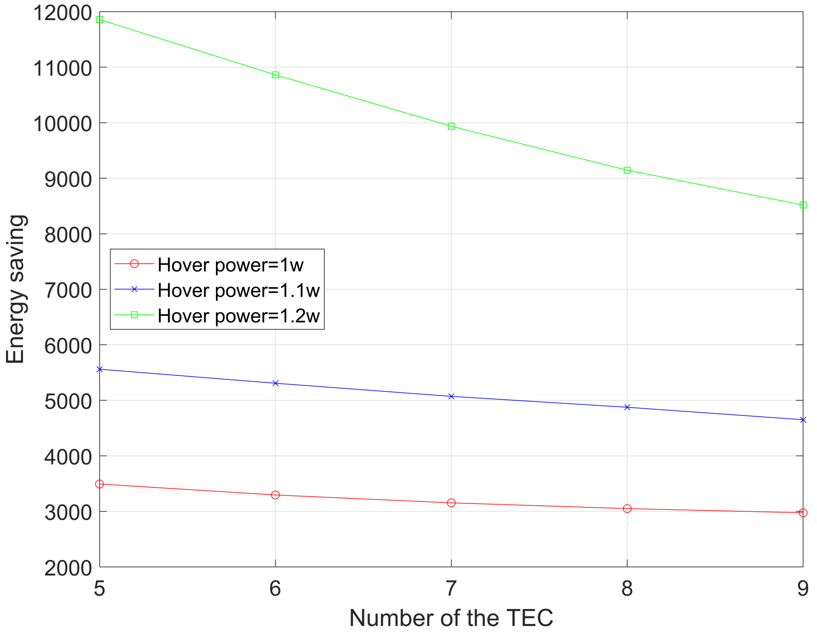


Figure 4.8: Energy saving of proposed GEDL in various scenarios.

In Figure 4.8, the energy savings in joules (J) at different amounts of TECs and different hover consumption power are shown. I set the training epoch is 200. I can see that in contrast to Figure 4.4, there is a declining trend in energy savings as the number of TECs increases. This is because accuracy revenue shows a logarithmic function trend. Providing more data when there are more TECs may increase energy savings, but not the corresponding accuracy gains. As a result, the total task size provided by TECs is decreasing and thus decreases the total energy saving. However, the GEDL I propose can always improve energy efficiency and save energy. Furthermore, the magnitude of the energy saving increase with the hover power increase is decreasing varies from Figure 4.4. It is likewise due to the existence of the trend in the logarithmic function of accuracy revenue. The decrease in regular task revenue due to time reduction makes the task size increase dramatically in order to reach the NE point.

Figure 4.9 illustrates the impact of changes in β value on energy saving. I evaluate

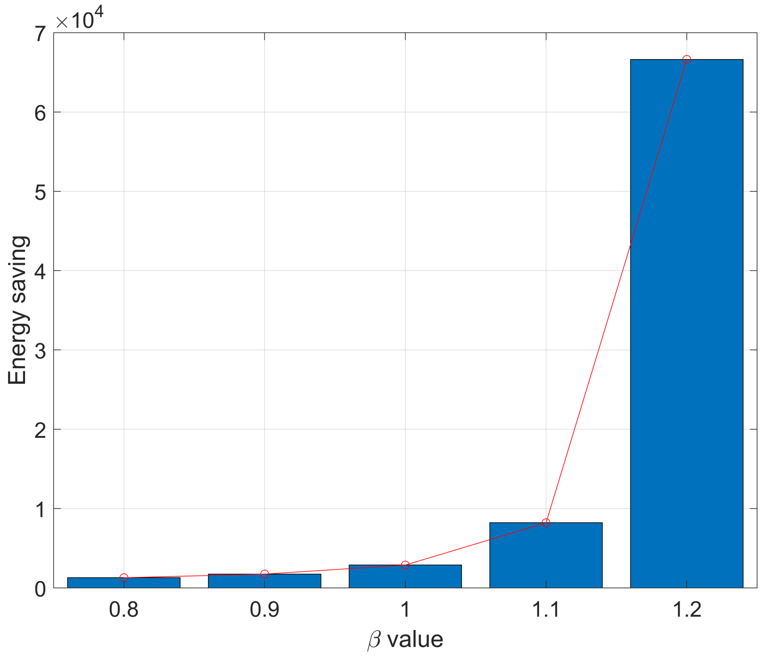


Figure 4.9: The impact of β value on energy saving.

this by adjusting the size of the energy cost monetary factor β. The smaller β means a higher energy cost price. I can observe that as the cost price grows, the overall energy saving of the network also rises exponentially. Due to the reduction in net income, the network members are more inclined to save energy for monetary benefits. Consequently, $m^t_{i,j}$ from the TEC i increases sharply in order to reach the NE point, thus making the energy saving increase.

4.6 Summaries

In this chapter, I first proposed a novel energy-efficient SemCom system in AENs. I then presented an EGTIM based on the Stackelberg game. In our EGTIM, the edge facilities on the terrestrial are incentive to transfer part of their semantic transmission tasks to the AEC via the traditional communication encoder. The AEC performs the semantic feature extraction of these tasks and transmits the semantic information to the subscribers. The energy efficiency of the aerial devices thus can be improved. In addition, I further proposed a GEDL framework based on the renewed EGTIM for energy-limited 3D networks updating semantic coders with Non-IID training data. The simulation results demonstrated the effectiveness of our mechanism and learning framework. In the next chapter, the optimisation mechanisms for one of the extended 6G networks, i.e., space-air-terrestrial networks, will be investigated.

Chapter 5

Space-air-terrestrial SemCom System

5.1 Introduction

In this chapter, the challenges of SemCom in SAT networks are analysed and investigated. In the SEC network, designing the SemCom system and updating the semantic coder presents several emerging challenges, e.g., mobility of SEC, low tolerance of service interruption and energy consumption, and privacy. Nevertheless, the existing distributed learning frameworks for SemCom in generic networks are however not automatically applicable to the SEC network. In addition, SemCom alters the transmission paradigm of SEC networks by increasing the computational load while reducing the communication load. Users are therefore required to develop optimal computational task strategies in case trained semantic coders are utilised for computation offloading. Such strategies need to be developed taking into account not only scenarios specific to SemCom in the SEC, but also operational factors that have not been considered in the existing SEC offloading research. Such factors include using both access modalities, the task processing entities, latency, energy consumption and privacy.

To tackle the above-mentioned challenges, in this chapter, I propose a SemCom system for SAT networks, i.e., SemCom SEC. In our proposed method, I split the SemCom service into in-maintenance (i.e., semantic coders need updating) and in-service (i.e., trained semantic coders are utilised for computation offloading) scenarios. For the in-maintenance scenario, I investigate real-time updating of deployed semantic coders in SemCom-SEC. A PSFed approach is then proposed to update semantic coders considering offloading QoS while privacy-preserving. For the in-service scenario, I study the computational task processing challenge of terrestrial users in the new SemCom paradigm. I then propose a new CTPS mechanism based on the Rubinstein bargaining game to minimise the users' processing delay and energy consumption while preserving users' privacy.

5.2 System model

In this section, the system model of the proposed SemCom-SEC is introduced. I then provide the computing, communication, path loss and semantic coder training model.

5.2.1 System description

Consider the SemCom-SEC (Figure 5.1), where terrestrial users are located in areas without having access to terrestrial edge service. Users can offload computation-intensive tasks to LEO satellite-borne edge facilities. In practice, an LEO satellite constellation is similar to a cellular network operating above the ground [66]. whereas the space cellular network is on the move, while ground users are relatively stationary.

I consider both types of approaches for users to access the SEC for computation offloading [36]. Users can communicate with LEO satellites directly through a C-band user-satellite radio link. Furthermore, they are also allowed to indirectly access the SEC through a TST via a C-band link to TST, and a Ka-band link between TST and SEC. The terrestrial C-band user-TST link spectrum resources are utilised

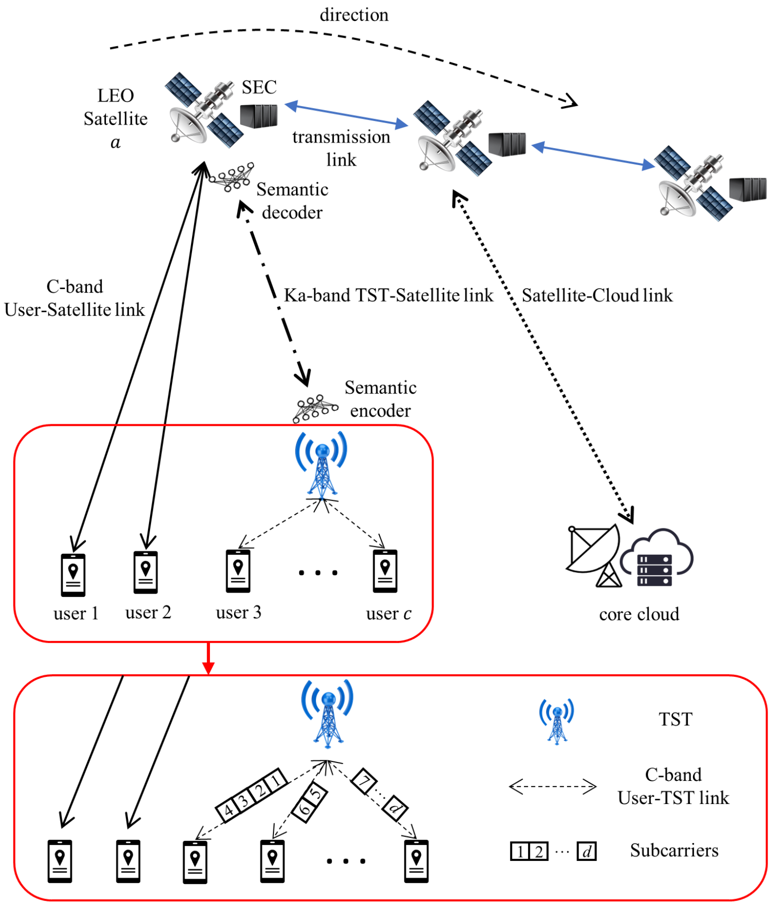


Figure 5.1: The proposed SemCom-SEC framework.

in an orthogonal frequency division multiple access (OFDMA) setting to optimise the utilisation of radio resources [38].

To improve the spectrum efficiency and QoS of SEC networks, semantic coders are deployed on the TSTs and LEO satellites for transmitting offloaded tasks over Ka-band. This is due to the mobility of the users and the fact that the offloading content is often variable and thus goal-oriented semantic coders need continuous updating. TSTs are primarily responsible for transmitting significant amounts of tasks to satellites and require extremely high spectral efficiency. Furthermore, their service area is fixed and the content to assist in task offloading (e.g., image recognition) only minimally varies. I consider factors such as utilisation, and reliability, for which goal-oriented SemCom is most appropriate for the TST-satellite link in SEC networks.

Moreover, due to the dynamic nature of the system and the limited storage resources of LEO satellites, it is not viable to store semantic decoders for all TSTs on the route. The semantic coders are therefore stored on the TST. Similarly, for economic and satellite storage resources considerations, at least the trained decoder of TSTs should be the same for the same transmission task. The TST delivers the related semantic decoders to the corresponding satellite when it needs to perform SemCom. Furthermore, LEO satellites can alternatively connect to the cloud servers on the terrestrial network via Ka-band backhaul links to provide cloud service for users.

In this model, a user may process indivisible computational tasks in either of the following five scenarios: 1) computing locally; 2) offloading the tasks to SEC over the user-satellite link; 3) offloading the tasks to the SEC via TST; 4) offloading the tasks to terrestrial cloud over the user-satellite link; 5) offloading the tasks to the terrestrial cloud via TST.

5.2.2 Computing models

Denote the set of LEO satellites as $\mathcal{A} = 1, 2, ..., a, ..., A$ and set of TSTs as $\mathcal{B} = 1, 2, ..., b, ..., B$. A TST b is on the terrestrial and provides service to C users within the coverage as a small cell in which the set of users in TST b's service range is denoted by $\mathcal{C} = 1, 2, ..., c, ..., C$. I consider each terrestrial user c to have indivisible computational sensitive tasks with the size in bits of $m_c \in m_1, m_2, ..., m_c, ..., m_C$, and the CPU cycles needed to execute one bit of tasks is δ. The local computation task latency of the user c can be given by

$$t_c^{LC} = \frac{\delta m_c}{f_c}, \tag{5.1}$$

where f_c is user c's CPU-cycle frequency with the unit cycles/s. The energy required to calculate locally is hence expressed as:

$$E_c^{LC} = p_c^{LC} t_c^{LC} = \varepsilon f_c^3 \frac{\delta m_c}{f_c} = \varepsilon \delta m_c f_c^2, \tag{5.2}$$

where $p_c^{LC} = \varepsilon f_c^3$ is the power needed to be computing locally and ε is the energy factor related to the electronics [45].

Similarly, if user c chooses to offload the tasks to SEC or the terrestrial cloud, the computational latency can be obtained by

$$t_c^{SEC} = \frac{\delta m_c}{f_a}, \tag{5.3}$$

$$t_c^{Cloud} = \frac{\delta m_c}{f_{Cloud}}, \tag{5.4}$$

where f_a and f_{Cloud} are the CPU-cycle frequency of the LEO satellite a being offloaded to and terrestrial cloud, respectively. Similar to [39] and [67], I assume that all LEO satellites have similar computing capabilities.

5.2.3 Communication models

There are two options for each user to access LEO satellites, i.e., directly access the LEO satellite or via a semantic encoder deployed on the TST. The total bandwidth of the C-band user-TST link is divided into D_0 orthogonal sub-carriers based on OFDMA manner. I have the transmission rate of the user c to the TST b on the subcarrier d_0 is

$$R_{c,d}^{cb} = B_{d_0}^{cb} \log_2(1 + \frac{p_{c,d_0}^{cb} g_{c,d_0}^{cb}}{\sigma_0^2}), \tag{5.5}$$

where $B_{d_0}^{cb}$, p_{c,d_0}^{cb} and g_{c,d_0}^{cb} are bandwidth, transmission power and the channel gain on sub-carrier d_0 in the user-TST link, separately. Further, in (5), σ_0^2 is the noise power in this link. Hence, the transmission delay from user c to TST b is

$$t_c^{cb} = m_c / \sum_{d_0=1}^{D_0} x_{d_0}^{cb} r_{c,d_0}^{cb}, \tag{5.6}$$

where $x_{d_0}^{cb} \in 0, 1$ is the allocation indicator of user-TST over the C-band. In the case of a sub-carrier d_0 in C-band is allocated to user c to offload the tasks, $x_{d_0}^{cb} = 1$; otherwise, $x_{d_0}^{cb} = 0$. Therefore, the transmission energy is

$$E_c^{cb} = t_c^{cb} \sum_{d_0=1}^{D_0} x_{d_0}^{cb} p_{c,d_0}^{cb}. \tag{5.7}$$

If user c chooses to access satellite a directly, due to the ultra-long propagation distance, the propagation delay is not negligible and the round-trip propagation delay is

$$t_c^{proa} = \frac{2h}{c_l}, \tag{5.8}$$

where h is the distance between user c and satellite a, c_l is the speed of light. I assume the users in the same TST, this TST and terrestrial cloud have the same distance to the satellite a.

Moreover, path loss should be considered when transmitting over long distances. I are not concentrating on the path loss in the user-TST link because they communicate in a small cell range and haven't got a significant impact on the transmission delay. The transmission rate from the user c to satellite a thus can be denoted by

$$R_c^{ca} = B_c^{ca} \log_2(1 + \frac{p_c^{ca} g_c^{ca}}{\sigma_0^2 PL_c^{ca}}), \tag{5.9}$$

where B_c^{ca}, PL_c^{ca}, p_c^{ca}, and g_c^{ca} are bandwidth, path loss, transmission power and the channel gain from the user c to satellite a, separately. Normally, the path loss PL for the satellite channels mainly consists of free-space path loss PL_f and atmospheric (rainfall) loss PL_r [68]. Hence, I assume the total path loss $PL = PL_f + PL_r$. I will specify these losses later. I then have the transmission delay and energy consumption when user c accesses the SEC a directly, which are given by

$$t_c^{ca} = \frac{m_c}{R_c^{ca}}, \tag{5.10}$$

$$E_c^{ca} = t_c^{ca} p_c^{ca}. \tag{5.11}$$

In contrast to users, the transmission process from TST b to satellite a integrates SemCom. It thus increases the computing delay while significantly decreasing the data required to be transmitted. The transmission rate of TST can be expressed as:

$$R_b^{ba} = B_b^{ba} \log_2(1 + \frac{p_b^{ba} g_b^{ba}}{\sigma_0^2 PL_b^{ba}}), \tag{5.12}$$

where B_b^{ba}, PL_b^{ba}, P_b^{ba} and g_b^{ba} are bandwidth, path loss, transmission power and the channel gain in TST b-satellite a link, respectively. In addition, since antennas of TSTs have good directivity, they can communicate with multiple LEO satellites via Ka-band and the corresponding interference can be ignored **10**, **22**, **23**. Therefore, the transmission delay of all users' tasks are transmitted from TST b to satellite a is

$$t_c^{ba} = \frac{\sum_{j=1}^{F} \psi m_j}{R_b^{ba}} + \frac{\sum_{j=1}^{F} m_j}{R_{SemCom}^{ba}}, \tag{5.13}$$

where F is the number of users allocated to offloading the task to satellite a and $F \in \mathcal{C}$. Furthermore, ψ is the compression ratio and the R_{SemCom}^{ba} is the rate of semantic extraction and semantic parsing, i.e., computing delay during data transmission.

Since the computation task calculation result is often much smaller than the offloaded data. I thus ignore the backhaul transmission delay links similar to [69] and [70]. Moreover, estimating the number of subcarriers provided by satellite a to user c is difficult due to a large number of satellite service users. I assume that the satellite transmits user data to the ground cloud with a constant transmission rate R_c^a similar to [36]. The transmission delay between satellite and cloud t_a^{Cloud} thus equals m_c/R_c^a. The propagation delay where user c chooses to offload to the terrestrial cloud is

$$t_c^{proC} = \frac{4h}{c_l}. \tag{5.14}$$

5.2.4 Path loss model

As mentioned previously, the path loss for the terrestrial-satellite channel is mainly free-space path loss PL_f and atmospheric (rainfall) loss PL_r. Free-space path loss is a basic power loss that increases depending on the communication distance. With the unit as dB, PL_f can be denoted by [71]

$$PL_f(dB) = 92.44 + 20\lg(h) + 20\lg(f), \tag{5.15}$$

where h is the communication distance unit in km, and f is the operating frequency with the unit of GHz.

Atmospheric loss is a type of signal absorption and scattering due to meteorological causes, i.e., mainly related to rainfall. The rain attenuation is described by [72]

$$PL_r(dB) = \xi L_E, \tag{5.16}$$

where ξ is the frequency-dependent parameter unit in dB/km and L_E is the effective path length unit in km. I first introduce the calculation method of ξ as:

$$\xi = k(R_{0.001})^v, \tag{5.17}$$

where $R_{0.001}$ is the rainfall rate, unit in mm/h. Further, k and v are coefficients given as:

$$k = [k_H + k_V + (k_H - k_V)cos^2(\omega)cos(2\tau)]/2, \tag{5.18}$$

$$v = [k_H v_H + k_V v_V + (k_H v_H - k_V v_V)cos^2(\omega)cos(2\tau)]/2, \tag{5.19}$$

where $\tau = \pi/4$ for circular polarization and ω is the elevation angle between terrestrial transmitter and satellite. Moreover, k_H, k_V, v_H, and v_V are coefficients related to operating frequency f and can be found out the specific value from [73]

L_E, is therefore

$$L_E = L_R v_{0.001}, \tag{5.20}$$

where L_R is the distance parameter related to rainfall height and $v_{0.001}$ is the adjustment factor. I have

$$v_{0.001} = \frac{1}{1 + \sqrt{sin(\omega)}(\frac{31(1-e^{-(\frac{\omega}{1+\chi})})\sqrt{LR^{\xi}}}{f^2} - 0.45)}, \tag{5.21}$$

where χ equals 36-—latitude— in the case of latitude less than 36^o, or equals 0. In most scenarios

$$L_R = \frac{h_R - h_s}{sin(\omega)} \tag{5.22}$$

where h_R is the rain height relative to the mean sea level and h_s is the altitude of the terrestrial transmitter, all units in km.

5.2.5 Semantic coder training model

In generally distributed learning frameworks based on FedAvg [30], the goal is to collaboratively train a global coder model among multiple TSTs while keeping TSTs' local data private. I set the $X_b = \{x_{in}^b\}_{b=1}^{s_b}$ as the data set of the TST b, where x_{in}^b is the in-th input sample and s_b is the size of the data set. The objective of FedAvg can be denoted by

$$\arg\min_{\Theta} \frac{1}{B} \sum_{b=1}^{B} L_b(\theta_b), \tag{5.23}$$

where θ_b is the coder model parameter of the TST b and $\Theta = \theta_1, \theta_2, ..., \theta_b$. Further, $L_b(\theta_b)$ is the loss function of the TST b trained by X_b. I utilise the mean squared error (MSE) loss as the loss function in this chapter. I have

$$L_b(\theta_b) = \frac{1}{s_b} \sum_{in=1}^{s_b} L_{MSE}(\theta_b; x_{b,in}, \widehat{x_{b,in}}), \tag{5.24}$$

where $\widehat{x_{b,in}}$ is the fitting output and L_{MSE} is the MSE loss.

5.3 Semantic coders: updating

Employing general FL frameworks for SemComs, TSTs need to upload encoder and decoder models to the SEC to implement federated aggregation after one communication round of training. The federated model then be sent back to TSTs for the next communication round of training. However, uploading and downloading all coder models by TSTs would cause long-term interruptions of the offloading-assisted service, significant energy consumption and lead to privacy leakage of entire coder models. I can express the general privacy leakage metric by [62]

$$\Theta_b(\theta_b) = \chi \log_2(1 + e^{1 - \frac{N_b + 1}{n_b}}), \tag{5.25}$$

where χ is the weight parameter, N_b is the total parameter number of the encoder model and n_b is the number of parameters transmitted. Since more training rounds and the more important parameters should have higher privacy sensitivity, I denoted the privacy leakage for TST b's encoder training by

$$\Theta_b(\theta_b) = \sum_{r=1}^{R} W_r \chi \log_2 (1 + e^{1 - \frac{\sum_i^{N_b} I_i n_{b,i} + 1}{\sum_i^{N_b} I_i n_{b,i}}}), \tag{5.26}$$

where r is the communication rounds and R is the total rounds. Thus, W_r is the model importance weight of training round r. Further, I_i is the parameter importance weight of transmitted parameter i.

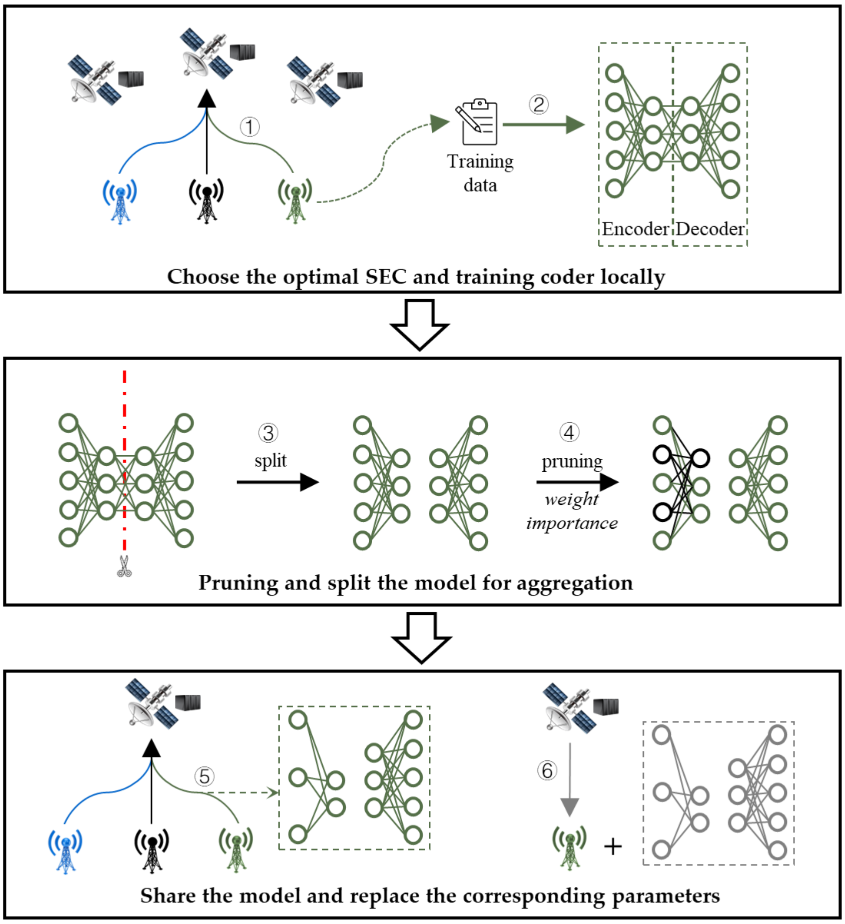

①
Training data
②
Encoder
Decoder
Choose the optimal SEC and training coder locally
③
split
④
pruning
weight importance
Pruning and split the model for aggregation
⑤
⑥
+
Share the model and replace the corresponding parameters

Figure 5.2: Framework of the proposed PSFed in one communication round.

In the proposed PSFed (Figure 5.2), the goal is to collaboratively train semantic coder models among multiple TSTs while reducing network service interruptions, and energy consumption, and decreasing the degree of privacy leakage. First, due to the high mobility of satellites, all TSTs are not always within the same satellite service area. TSTs are required to select the most appropriate satellite for each model aggregation round from the multiple satellites based on real-time circumstances. Taking into account training delay and energy consumption jointly, the selection algorithm can be denoted by

$$\min_{x_a} \sum_{a=1}^{A} x_a(\alpha \max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\} + \sum_{b=1}^{B} \beta p_b^{ba} \frac{M_{b,r}}{R_b^{ba}}), \tag{5.27a}$$

$$s.t. \sum_{a=1}^{A} x_a = 1, \forall b \tag{5.27b}$$

$$x_a = \{0, 1\}, \tag{5.27c}$$

$$\sum_{r=1}^{R} \frac{M_{b,r}}{R_b^{ba}} \leq t_b^{'}, \forall b \tag{5.27d}$$

$$\max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\} < t_a^{'}, \forall a \tag{5.27e}$$

where $\max\{\frac{M_{b,r}}{R_b^{ba}} + \frac{2h^{ba}}{c_l}|b \in \mathcal{B}\}$ is the training transmission and propagation delay, identified by the TST with the longest transmission and propagation time. Here, A is the number of accessible satellites of all TSTs, and h^{ba} is the distance between TST b and satellite a. Further, $\sum_{b=1}^{B} \beta p_b^{ba} \frac{M_{b,r}}{R_b^{ba}}$ is the total energy consumption of transmission from TSTs to a satellite. In (5.27a), α and β are weight parameters to balance the importance and unit of latency and energy consumption. Furthermore, p_b^{ba} is the transmission power of TST b to satellite a, and x_a is the federated decision for all TSTs. Constraint (5.27d) ensures that the transmission time of the TST for training the semantic model remains less than the maximum tolerable service interruption time. Also, $M_{b,r}$ is the coder model size in communication round r, $t_b^{'}$ is the maximum tolerable service interruption time and $t_a^{'}$ is the maximum service time of the satellite a in this region. The optimization problem in (5.27) is a simple 0,1 linear programming and hence can be easily solved.

During training in each communication round, I split the coder model into an encoder and a decoder. Only the decoder model needs entire federated aggregation. This is due to LEO satellites having limited storage capacity, it is not practical to use individual decoder models for each task of each TST. For economic considerations, I argue that TSTs require a shared decoder model to be used. I then encourage TSTs to assess the importance of the encoder parameters during the local training phase. Inspired by continual learning [74], changes in parameters with different importance have a different impact on the output results. I thus evaluate parameter importance according to the implications of parameter changes on the loss function. I express the change in the loss by

$$L_b(\theta_b + \delta) - L_b(\theta_b) \approx \sum_{i=1}^{s_b} g_{b,i}\delta_{b,i}, \tag{5.1}$$

where g_i is the gradient and δ_i is the update of parameter i during this parameter assessment period of the TST b. Setting $g_i = \frac{\partial L_b}{\partial \theta_{b,i}}$ during online training, the parameter importance weight is

$$I_i = -\frac{\partial L_b}{\partial \theta_{b,i}}\delta_{b,i}. \tag{5.2}$$

Subsequently, to reduce the training communication cost, I prune the encoder models uploaded by TSTs according to parameter importance. Parameters with high importance contain most of the valid information [75] and therefore can provide further valid information to the aggregated model than lower-important parameters. The lower-importance parameters are thus encouraged to be pruned. The pruning here differs from the conventional ML studies. It is not the deletion of the training model parameters, but the non-transmission of the pruned parameters for federated aggregation. The corresponding SEC generates a global encoder model and a global decoder model based on the federated aggregation of the number of the received parameters. Once TST receives the global decoder model and personalised pruned global encoder model, it merely substitutes the local decoder and substitutes important parameters of the local encoder. It trains the individual local coder again based on the personal encoder model and the global decoder model in the next communication round of training.

Furthermore, the closer to the completion of the training, the higher the importance of the parameters. To further reduce the privacy leakage degree, our proposed PSFed progressively increases the pruning ratio according to the number of communication rounds. This is until the coder model is split and only the decoder model is federated aggregated. The more important privacy training models are thus kept local.

The objective of PSFed during training thus is denoted by

$$\arg\ \min_{\Theta, Y} \sum_{b=1}^{B} L_b(y_b^1 \theta_{b,1}, y_b^2 \theta_{b,2}, ..., y_b^n \theta_{b,N_b}), \tag{5.3a}$$

$$s.t. \quad \sum_{r=1}^{R} \frac{M_{b,r}}{R^{ba}} \leq t_b^{'}, \forall b \tag{5.3b}$$

$$\sum_{r=1}^{R} W_r \chi \log_2(1 + e^{1 - \frac{\Sigma_i^{N_b} I_i n_{b,i} + 1}{\Sigma_i^{N_b} I_i n_{b,i}}}) \leq \Theta_b^{'}, \forall b \tag{5.3c}$$

where $y_b^n \in [0, 1)$ is the aggregation weight vector of parameter i in TST b and $Y = y_1, y_2, ..., y_b$. Further, $M_{b,r}$ is the coder model size in r communication round and $t_b^{'}$ and $\Theta_b^{'}$ are the maximum tolerable service interruption time and privacy leakage, respectively. The procedure of the PSFed is demonstrated in Algorithm 5.1.

5.4 Semantic coders: in service

In this section, the problem of users' computational task processing schedule for SemCom-SEC is presented first. I then detail the proposed CTPS.

5.4.1 Problem for computational task processing

In service offloading decision-making, I consider the SemCom-SEC with C users severed by one TST b in A satellite coverage. Each user has five task processing choices, 1) local computing; 2) offloading the tasks to SEC directly; 3) offloading the tasks to SEC via the TST; 4) offloading the tasks to the terrestrial cloud directly; 5) offloading the tasks to the terrestrial cloud via the TST. I firstly list the user c's cost functions in terms of processing delay and energy consumption for each option

Algorithm 5.1 PSFed

```
Input: dataset $\{X_1, X_2, ..., X_b\}$, model size $\{M_1,M_2,...,M_b\}$ and total communication rounds $R$
Output: trained coder models $\{\theta_1, \theta_2, ..., \theta_b\}$
Initialize: the TSTs' model parameters and the importance weight of parameters
SECs:
for each communication round $r \in R$ ::
  $Y_b^{r+1}, \theta_b^{r+1} \longleftarrow TST\ update(\theta_b^r)$
  Update $\{\theta_{b,1}, \theta_{b,2}, ..., \theta_{b,N_b}\}$ according to $Y_b^{r+1}$ and $\theta_b^{r+1}$
end for
TSTs:
TST $b$ receives $\theta_b$ from the SEC
TSTs choose the optimal SEC for federated aggregation
for each TST in parallel:
  for each local training epoch:
    Loss $\longleftarrow = \frac{1}{s_b} \sum_{in=1}^{s_b} L_{MSE}(\theta_b; x_{b,in}, \widehat{x_{b,in}})$
  end for
  foreach encoder parameter $i$:
    $I_i = -\frac{\partial L_b}{\partial \theta_{b,i}} \delta_{b,i}$
  end for
  Splitting coder model and pruning encoder model based on $I_i$ in the case of satisfying:
  $\begin{cases} \sum_{r=1}^{R} \frac{M_{b,r}}{R^{ba}} + \frac{M_{b,r}}{R^{ab}} \leq t_b^{'} \\ \Theta_b(\theta_b^r) \leq \Theta_b^{'} \end{cases}$
  Obtain $\theta_b^r$ to be shared
  return: $\theta_b^r$
end for
```

in order as follows based on section 2:

$$\Phi_{c1} = \alpha t_c^{LC} + \beta E_c^{LC}, \tag{5.1}$$

$$\Phi_{c2} = \alpha(t_c^{proa} + t_c^{ca} + t_c^{SEC}) + \beta E_c^{ca}, \tag{5.2}$$

$$\Phi_{c3} = \alpha(t_c^{proa} + t_c^{cb} + t_c^{ba} + t_c^{SEC}) + \beta E_c^{cb}, \tag{5.3}$$

$$\Phi_{c4} = \alpha(t_c^{proa} + t_c^{ca} + t_c^{Cloud} + t_a^{Cloud}) + \beta E_c^{ca}, \tag{5.4}$$

$$\Phi_{c5} = \alpha(t_c^{proC} + t_c^{cb} + t_c^{ba} + t_c^{Cloud} + t_a^{Cloud}) + \beta E_c^{cb}, \tag{5.5}$$

where Φ_c is the actual processing cost when the user c sizing a task. It is related to user task processing decisions, the transmission power, and the number of subcarriers allocated. In the above, t_a^{Cloud} is the transmission delay between satellite and cloud as mentioned in Section II-C. We also utilise $\gamma_{ic} = \{0, 1\}$ to represent the offloading decision of user c and $\gamma_{ic} \in \{\gamma_{1c}, \gamma_{2c}, \gamma_{3c}, \gamma_{4c}\}$. If user c chooses one processing strategy, the indicator for the corresponding strategy equals 1, otherwise equals 0. We argue that the optimal decision for a user is to minimise the latency and energy consumption of the processing tasks. Mathematically, the optimisation task processing strategy problem of user c thus can be formulated as a MINLP problem:

$$\begin{aligned}
\min_{\gamma_c, f_c, p^{cb}_{c,d_0}, m_{c,d_0}, p^{ca}_c} \quad & \sum_{a=1}^{A} \Phi_c = (1 - \gamma_{1c} - \gamma_{2c} - \gamma_{3c} - \gamma_{4c})\Phi_1 \\
& + \gamma_{1c}\Phi_{c2} + \gamma_{2c}\Phi_{c3} + \gamma_{3c}\Phi_{c4} + \gamma_{4c}\Phi_{c5}, && \text{(5.6a)} \\
s.t. \quad & f_{cloud} \geq f_a \geq f_{c,max} \geq 0, && \text{(5.6b)} \\
& \gamma_{1c}, \gamma_{2c}, \gamma_{3c}, \gamma_{4c} \in \{0, 1\}, && \text{(5.6c)} \\
& \gamma_{1c} + \gamma_{2c} + \gamma_{3c} + \gamma_{4c} \leq 1, && \text{(5.6d)} \\
& \sum_{d_0=1}^{D_0} x^{cb}_{d_0} x^{cb}_{c,d_0} \leq P_{c,max}, && \text{(5.6e)} \\
& P^{ca}_c \leq P_{c,max}, && \text{(5.6f)} \\
& x^{cb}_{d_0} \in \{0, 1\}, && \text{(5.6g)} \\
& \sum_{d_0=1}^{D_0} x^{cb}_{d_0} \leq D_0. && \text{(5.6h)}
\end{aligned}$$

The constraint (5.36b) guarantees that edge and cloud have strong computing capability that is not less than users' maximum computing capability $f_{c,max}$. Constraints (5.36c) and (5.36d) show the relationship between $\gamma_{1c}, \gamma_{2c}, \gamma_{3c}$and γ_{4c}. In constraints (5.36e) and (5.36f), $P_{c,max}$ is the maximum available transmission power of user c to TSTs or satellites. The constraint (5.36g) denotes the subcarrier allocation indicator. The constraint (5.36h) means that the number of allocated subcarriers should not exceed the total number of sub-carriers.

However, this is an MINLP problem with incomplete information due to privacy concerns. This is because users need the allocation of subcarriers to make decisions. Nevertheless, such information is relevant to decisions and privacy information (local computing capability and transmission power etc.) from other users. This MINLP problem thus is computationally complex and hard to be solved.

5.4.2 CTPS

In this chapter, I propose a CTPS mechanism (Figure 5.3) to minimise the delay and energy consumption of users to process computational tasks, while privacy-preserving and equitable. It is divided into two steps. Firstly, it converts the optimisation task processing strategy problem with privacy considerations into a complete information problem based on the Rubinstein bargaining model [76] equitably. Subsequently, users develop the optimisation task processing strategies by solving the complete information MINLP problem of Eq. (5.36). I detail our CTPS mechanism as follows.

5.4.3 First step of the CTPS mechanism

I enable users to communicate/bargain with TST several times so that subcarriers are allocated fairly without privacy leakage based on the Rubinstein bargaining game. TST acts as the bidder and the user has the option to continue the game or leave the game.

The gaming process is limited to two periods. In the first period, the users send the offloading request to the TST. Upon receiving users' offloading requests, without loss of generality and fairness, TST allocates the number of C-band subcarriers based on the size of the tasks offloaded by users. Further, the transmission delay of the TST to the satellite and semantic extraction delay are also notified via this communication. To achieve the game-perfect equilibrium, the cost function for user c to assess to continue participating in the game can be denoted by

$$\mu_c^{'} = \epsilon\iota\Phi_c^{'},\ \Phi_c^{'} = \{\Phi_{c3}, \Phi_{c5}\}, \tag{5.1}$$

where $\iota \in (0,1)$ is the bargaining discount factor that represents the revenue loss value for the second-period communication due to the bargaining process being time and energy-consuming. Further, $\epsilon \geq 1$ is the weight parameter to evaluate the further possible benefit by applying offloading again via the TST b, i.e., remaining engaged in the game. This is attributable to some users abandoning their requests for TST offloading due to not being allocated a satisfactory number of C-band subcarriers. The actual number of subscribers should eventually be greater than or equal to this allocation.

Simultaneously, the strategies of various users also affect the user-satellite link interference for different users. In order to estimate the influence of interference, pricing is a frequently utilised method in the game theory employed studies [77]. I hence rewrite the part of the cost function for user c considering interference pricing as:

$$\mu_c^{''} = \Phi_c^{''} + \alpha \varrho m_c \varpi, \ \Phi_c^{''} = \{\Phi_{c2}, \Phi_{c4}\}, \tag{5.2}$$

where ϱ is the factor for the interference related to user number, transmission power and channel gain etc. Further, $\varpi \in [0, 1]$ is the proportion to denote the anticipation rate of not performing local computing users, thus predicting the interference time suffered.

Finally, the incomplete information MINLP problem is converted to a complete information MINLP problem. Users thus could develop the optimal processing decision based on allocated subcarriers and the calculation frequency or transmitting power in the second step.

5.4.4 Second step of the CTPS mechanism

In the second step, users make the decision based on the complete information MINLP problem of Eq. (5.36) to minimise the latency and energy consumption of the processing tasks. As the maximum number of satellites expected to be accessible at the same time is extremely limited [78], the decision problem Eq. (5.36) can be considered as $5 \cdot A$ independent subproblems. In case of the local computing, the best user c's CPU-cycle frequency f_c is only related to local computing costs. I thus

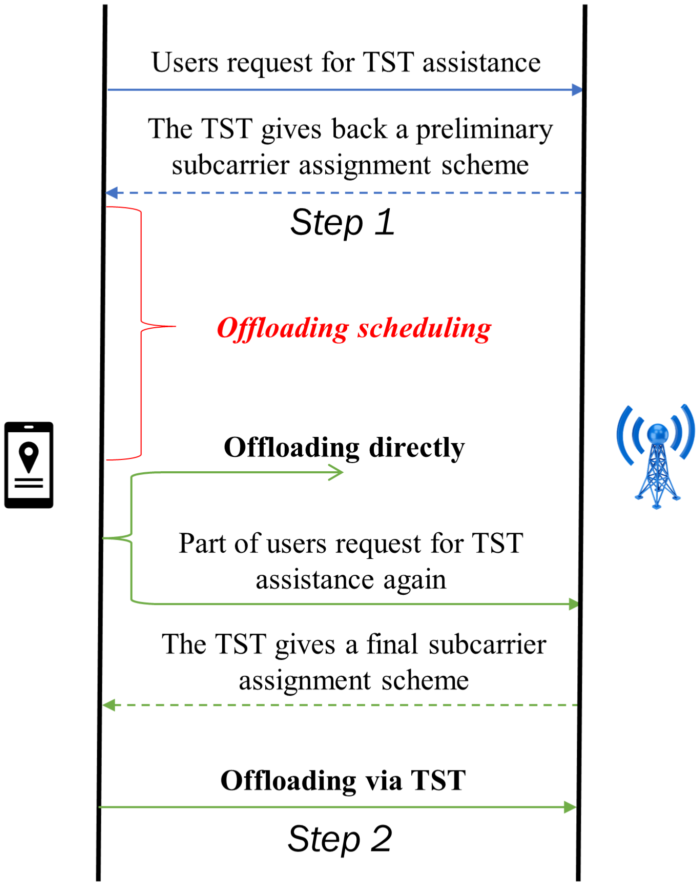

Users request for TST assistance
The TST gives back a preliminary subcarrier assignment scheme
Step 1
Offloading scheduling
Offloading directly
Part of users request for TST assistance again
The TST gives a final subcarrier assignment scheme
Offloading via TST
Step 2

Figure 5.3: Proposed CTPS mechanism.

can express the f_c optimisation subproblem as:

$$\min_{f_c} \quad \Phi_{c1} = \alpha\frac{\delta m_c}{f_c} + \beta\varepsilon\delta m_c f_c^2, \tag{5.3a}$$

$$s.t. \quad (36b). \tag{5.3b}$$

The above subproblem (5.39) has convex objective functions and constraints. Therefore, subproblem (5.39) can be solved by standard convex optimisation approaches promptly. In addition, in case the user needs to employ TSTs, the user needs to derive the optimal subcarrier task allocation strategy m_{c,d_0} and subcarrier transmission power p_{c,d_0}^{cb}. To model and optimise the transmission power, in CTPS, I assume each subcarrier in the same link accomplishes the transmission tasks at the same time for fully using spectrum resources in a synchronous manner based on previous studies [69], [79]. As the allocated subcarrier for user c is known, I set η to denote the number of allocated subcarriers. I can simplify the optimisation problem associated with TST as:

$$\min_{m_{c,d_0},p^{cb}_{c,d_0}} \quad \sum_{d_0=1}^{D_0}(\frac{\alpha x^{cb}_{d_0} m_{c,d_0}}{\eta r^{cb}_{c,d_0}} + \frac{\beta p^{cb}_{c,d_0} x^{cb}_{d_0} m_{c,d_0}}{r^{cb}_{c,d_0}}), \tag{5.1a}$$

$$s.t. \quad (36e),(36g),(36h), \tag{5.1b}$$

$$\sum_{d_0=1}^{D_0} x^{cb}_{d_0} m_{c,d_0} = m_c. \tag{5.1c}$$

I only need to consider the situation that $x^{cb}_{d_0} = 1$. By relaxing constraints, I have the Lagrangian function for Eq. (5.40a) as:

$$\begin{aligned} L = &\sum_{d_0=1}^{D_0} x^{cb}_{d_0}(\frac{\alpha m_{c,d_0}}{\eta r^{cb}_{c,d_0}} + \frac{\beta p^{cb}_{c,d_0} m_{c,d_0}}{r^{cb}_{c,d_0}}) \\ &+ \varphi(\sum_{d_0=1}^{D_0} x^{cb}_{d_0} p^{cb}_{c,d_0} - P_{c,max}) + \lambda(m_c - \sum_{d_0=1}^{D_0} x^{cb}_{d_0} m_{c,d_0}), \end{aligned} \tag{5.1}$$

where φ and λ are the Lagrangian multipliers. The dual function thus is $\min_{m_{c,d_0},p^{cb}_{c,d_0}} L$. Then, I can observe that Eq. (5.41) can be further decomposed into D_0 independent subproblems, and the actual objective function in each d_0 subproblem can be denoted by

$$\min_{m_{c,d_0},p^{cb}_{c,d_0}} L_{d_0} = \frac{\alpha m_{c,d_0}}{\eta r^{cb}_{c,d_0}} + \frac{\beta p^{cb}_{c,d_0} m_{c,d_0}}{r^{cb}_{c,d_0}} + \varphi p^{cb}_{c,d_0} + \lambda m_{c,d_0}. \tag{5.2}$$

For simplicity, I define

$$H_{d_0} = \frac{\alpha}{\eta r^{cb}_{c,d_0}} + \frac{\beta p^{cb}_{c,d_0}}{r^{cb}_{c,d_0}}. \tag{5.3}$$

According to Karush-Kuhn-Tucker conditions, taking the partial derivatives of L_{d_0} with respect to p^{cb}_{c,d_0} and m_{c,d_0}, respectively. I have

$$
\begin{cases}
\dfrac{\partial L_{d_0}}{\partial p^{cb}_{c,d_0}} = m_{c,d_0}\dfrac{\partial H_{d_0}}{\partial p^{cb}_{c,d_0}} + \varphi = 0 & \text{(5.4a)}\\
\dfrac{\partial L_{d_0}}{\partial m_{c,d_0}} = H_{d_0} - \lambda = 0 & \text{(5.4b)}\\
\varphi(\sum\limits_{d_0=1}^{D_0} x^{cb}_{d_0}p^{cb}_{c,d_0} - P_{c,max}) = 0. & \text{(5.4c)}
\end{cases}
$$

Thus, I have

$$
\begin{cases}
\varphi = 0, \sum\limits_{d_0=1}^{D_0} x^{cb}_{d_0}p^{cb}_{c,d_0} \leq P_{c,max}, & \text{(5.1a)}\\
\varphi > 0, \sum\limits_{d_0=1}^{D_0} x^{cb}_{d_0}p^{cb}_{c,d_0} = P_{c,max}. & \text{(5.1b)}
\end{cases}
$$

In case of Eq. (5.45a), the p^{cb}_{c,d_0} can be directly solved by Eq. (5.44) causing $m_{c,d_0} \neq 0$. After deriving the optimal p^{cb}_{c,d_0}, m_{c,d_0} can be easily solved as all subcarriers have the same subcarrier completion time. Only if the solution $\sum_{d_0=1}^{D_0} p^{cb}_{c,d_0} = P_{c,max}$, I need to consider Eq. (5.45b). In that case, the Lagrangian multipliers can be obtained by the sub-gradient method and further achieve the optimal p^{cb}_{c,d_0}, m_{c,d_0}. Moreover, as I utilise the Lagrangian dual decomposition method, the solution may exist a duality gap. However, this gap should approach zero and can be ignored in practical systems as the number of subcarriers D_0 is large enough [38].

Therefore, users can make the decision based on the computation cost of various alternatives, without compromising privacy. Throughout the CTPS, the user is only communicated externally about the size of the tasks being processed. It also needs to be known by TST during the offloading process. Hence the CTPS protect the privacy of computing power, transmit power, etc. The CTPS and offloading decision process is summarised as Algorithm 5.2.

Algorithm 5.2 CTPS

Input: Tasks m_c generation

Output: The computation offloading and resource allocation result $\gamma_c, f_C, p^{cb}_{c,d_0}, m_{c,d_0}, x^{cb}_{d_0}$

```
Initialize the optimal TST transmission power $p^{ba}_b$
Obtain necessary information $x^{cb}_{d_0}$ after first period game
Obtain the necessary information $x^{cb}_{d_0}$ after first period game
Calculate optimally $f_c$
Relax Eq. (5.40)
if $\varphi = 0$:
  $p^{cb}_{c,d_0} \longleftarrow \frac{\partial H_{d_0}}{\partial p^{cb}_{c,d_0}}$
  $m_{c,d_0} = \frac{m_c p^{cb}_{c,d_0}}{\sum_{d_0=1}^{D_0} x^{cb}_{d_0} p^{cb}_{c,d_0}}$
else:
  $p^{cb}_{c,d_0} \longleftarrow$ Eq. (5.44)
  $m_{c,d_0} = \frac{m_c p^{cb}_{c,d_0}}{P_{c,max}}$
end if
Find the maximum $\Phi_c$ and derive $\gamma_c$
if $\gamma_{c3} + \gamma_{c5} = 1$:
  Obtain the necessary information $x^{cb}_{d_0}$ after the second period game
  Obtain updated $p^{cb}_{c,d_0}$ and $m^{cb}_{c,d_0}$
end if
Find the maximum $\Phi_c$ and derive $\gamma_c$
```

5.5 Simulation results

5.5.1 Simulation setting

In this section, I evaluate the performance of the present PSFed and CTPS. In the simulations, if not specifically mentioned, I set the parameters as follows. The LEO satellites' coverage radius is 280 km and the vertical altitude is 780km based on the Iridium satellite system [80]. The frequencies of the C-band and the Ka-band are 4.5 GHz and 30 GHz separately based on 3GPP specifications [81]. I assume the number of C-band subcarriers is 128, the maximum transmission power of users is 23 dBm and the transmit power of each TST is 30 dBm [38]. The offloading task is assumed an image recognition task and the semantic coder is considered an autoencoder based on the convolutional autoencoder (CAE) similar to [17]. Communication rounds for the proposed PSFed to aggregate the semantic encoder are 20 rounds. The coder settings are listed in Table 5.1. Furthermore, I set the number of CPU cycles for computing one bit δ as 120 cycles/bit, which is from the real applications [45]. I assume all users have the same CPU frequency f_c, and set it as 0.5×10^9 cycles/s. The computation capabilities of SEC on satellite a and the cloud server are 3×10^9 cycles/s and 10×10^9 cycles/s, respectively [39]. Moreover, I assume weight parameters of latency and energy consumption are set as $\alpha = 0.5$ and $\beta = 0.5$, and weight parameters in bargain process ι and ϵ are all considered as 1. In addition, the atmospheric loss is adopted, and the related coefficients are shown in Table 5.2 [73]. The simulation parameters are also listed in Table 5.3.

Table 5.1: The setting of the CAE

Encoder	**Neuron num**	**Decoder**	**Neuron num**
Conv+ReLU	512	transConv+ReLU	10
Conv+ReLU	256	transConv+ReLU	32
Conv+ReLU	128	transConv+ReLU	64
Conv+ReLU	64	transConv+ReLU	128
Conv+ReLU	32	transConv+ReLU	256
Conv+Sigmod	10	transConv+Sigmod	512

Table 5.2: Rainfall coefficients

C-band	Value	Ka-band	Value
k_H	0.0001340	k_H	0.2403
k_V	0.0002347	k_V	0.2291
v_H	1.6948	v_H	0.9485
v_V	1.3987	v_V	0.9129

Table 5.3: Simulation parameters

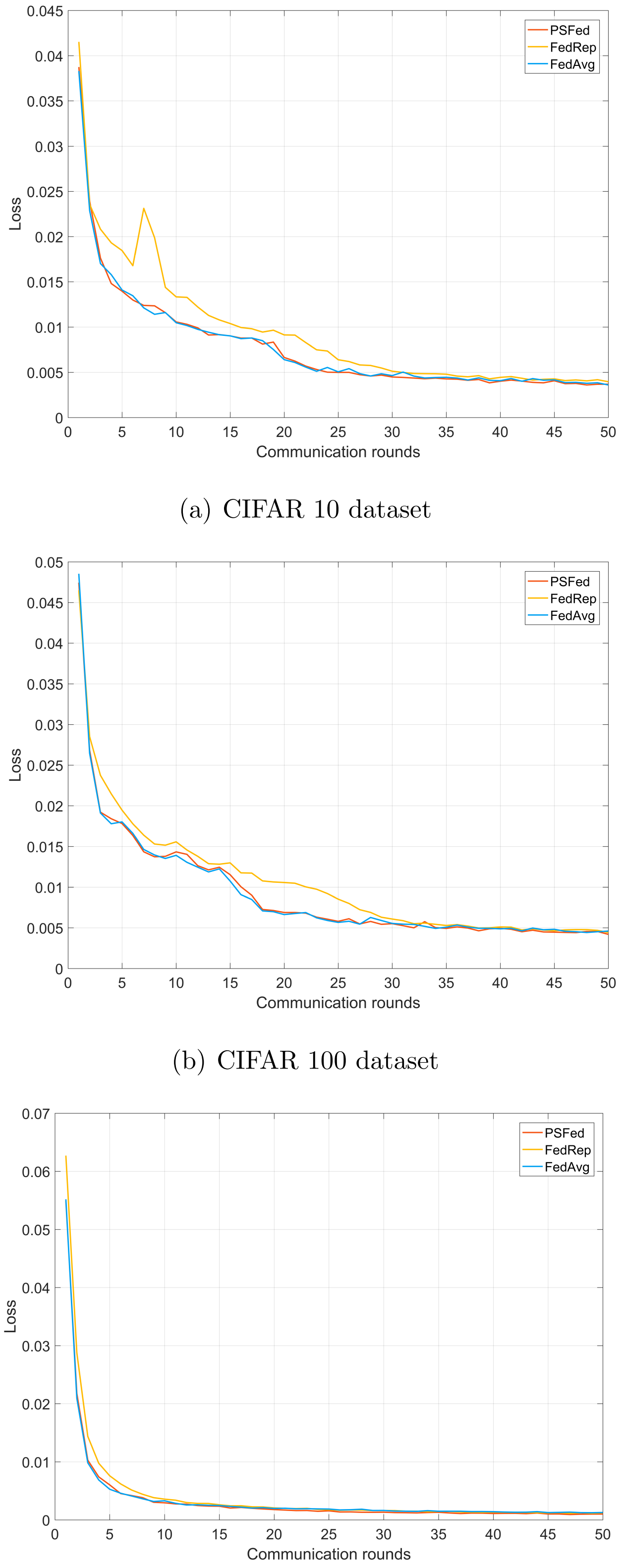
Parameters	Default values
The coverage radius of LEO satellites	280 km
Ka-band carrier frequency	30 GHZ
C-band carrier frequency	4.5GHZ
Number of C-band subcarriers	128
The maximum transmit power of each user	23dBm
Transmit power of TST	30 dBm
h	780km
δ	120
ε	10^{-26}
f_c	0.5×10^9 cycles/s
f_a	3×10^9 cycles/s
f_{Cloud}	10×10^9 cycles/s
α, β	0.5
ι, ϵ	1

5.5.2 Performance evaluation of the proposed PSFed

Figure 5.4 illustrates the convergence speed of the different frameworks under different transmission tasks. The TSTs' images are from CIFAR 10 [54], CIFAR 100 [54] and MNIST [82] image datasets and TSTs perform federated aggregation after every five local epochs. Based on the feasibility in SEC networks, I compare the proposed PSFed with the generalised learning approach for SemCom [11], [12], i.e., FL frameworks based on the FedAvg [30]. Further, based on the existing FL methods that are potentially for SEC SemCom, the federated decoder-only method, i.e., FL based on FedRep [83], is also compared to demonstrate the effectiveness of our PSFed. Moreover, I set the training sample to 5000 images per TST to reflect the differences between the frameworks more effectively. It can be observed that our PSFed achieves similar convergence rates to the FedAvg and is much better than the FedRep, regardless of the dataset. This is because our method aggregates important weights in the early stages of training and therefore accelerates convergence similarly to the FedAvg with all parameters aggregated.

In Figure 5.5, I compare the total communication cost of PSFed, FedRep and FedAvg during training. I assume that each neuron transmitted consumes the same amount of communication resources. The communication cost is therefore defined as the number of neurons transmitted during communication. It is seen that the PSFed expenses are approximately the same communication cost as the FedAvg in the early stages of training. The growth then gradually slows down and increases at the same magnitude as the FedAvg after round 20. This is because the PSFed gradually decreases the number of weights aggregated by the encoder model. In round 20, the number of aggregated weights for the encoder model is 0, the same as the FedRep, only the decoder model is aggregated. Therefore, the PSFed only consumes additional communication resources for the importance weight aggregation than the FedRep. Considering that the FedRep converges much more slowly than the proposed PSFed, the total communication resource consumption can be considered to be similar. However, in comparison to the FedAvg, the communication consumption of our PSFed decreases by 40.50% in round 50.

I evaluate the total model privacy leakage during training in Figure 5.6 according to Eq. (5.26). I assume that the model in each communication round has the same importance and that each neuron is of equal importance. It can be observed that



(a) CIFAR 10 dataset

(b) CIFAR 100 dataset

(c) MNIST dataset

Figure 5.4: Convergence speed of various learning algorithms with different datasets.

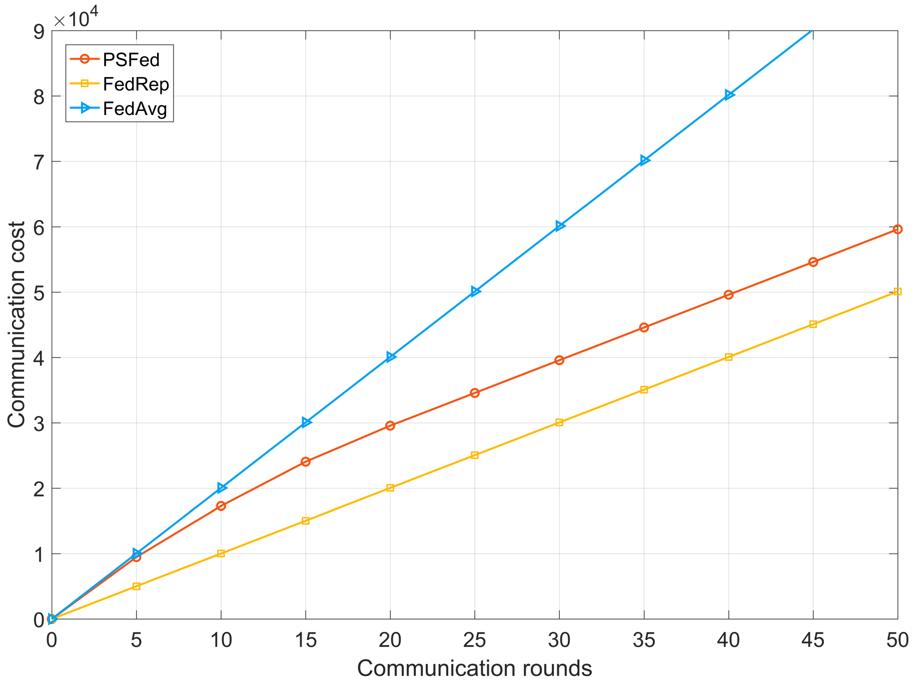


Figure 5.5: Communication cost of various learning approaches.

PSFed is initially similar to FedAvg leakage and subsequently follows the same growth trend as FedRep. This is equally due to the number of PSFed decreasing importance weight aggregations. After training, both the PSFed and the FedRep encoder models are saved locally. It is foreseeable that if the importance of each round of communication changes, the PSFed would be extremely close to the FedRep in terms of total privacy leakage. In addition, the privacy leakage of PSFed should widen the gap with FedAvg, even though the privacy leakage of our PSFed already decreases by 51.43% in round 50 in comparison to FedAvg in the same importance.

In Figure 5.7, the accuracy of the different frameworks under different transmission tasks is shown. I evaluate the accuracy utilising Peak Signal-to-Noise Ratio (PSNR), a general metric for evaluating image transmission in SemCom [17]. I have

$$PSNR = 10\lg\frac{MAX^2}{MSE}(dB), \tag{5.1}$$

where MAX is the maximum value for a pixel and MSE is the mean squared deviation. Since different datasets have different MAX, I assume that the learning method with the smaller MSE has a higher accuracy. It is seen that the FedRep is significantly the least accurate with different datasets trained. The accuracy of PSFed is similar to FedAvg but slightly higher. Because encoder models of both PSFed and FedRep are kept at the TST that are not aggregated when training is completed. Some aggregation information thus is lacking. However, the average training accuracy of the PSfed decreased by only 0.33% relative to the FedAvg due to the important weight aggregation acting as pre-training. Compared to the FedAvg, the accuracy loss of the PSfed deems acceptable given the significant communication cost and privacy concerns of the former.

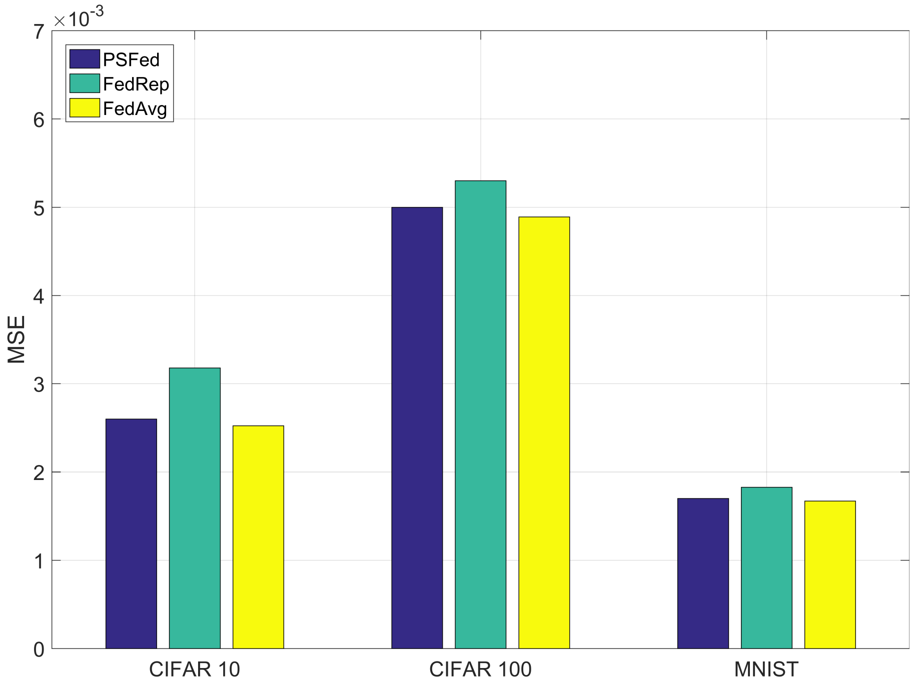


Figure 5.6: Accuracy of various learning algorithms with different datasets.

5.5.3 Performance evaluation of the proposed CTPS

Figure 5.8 illustrates the impact of users in one TST coverage on the total cost. As users are not always able to offload tasks via the TST, the proposed CTPS is compared with the local computing, offloading to the SEC directly, offloading to the cloud directly and CTPS without the game. The task size for each user is randomly generated over a range of 5 kb-300kb and subjected to 50 times replications of the simulation. From the figure, the total cost grows with the number of users. This is because raising the number of users increases the corresponding number of computing tasks and thus the total cost of users. Moreover, the total cost of the proposed CTPS always keeps the total cost to the minimum and the advantage increases as the number of users increases. Because the reallocation of resources through our design game scheme increases the efficiency of network resource utilisation.

In Figure 5.9, I show the offloading and computing cost of a single user versus the size of generating tasks. It is observed that the cost increases with the data size for all schemes. Our proposed mechanism always has a lower cost compared to the other three approaches. In case of the data size is small (10 kb), our CPTS choose local computing as the optimal option. As the data size grows, the local computing latency and energy consumption increase, and CTPS chooses other minimum cost strategies, i.e., offload tasks to the SEC via the TST. After 250kb, the optimal value of our mechanism fluctuates. This is due to the data size being large enough, and the best strategy changes to offload tasks to the cloud via a TST. Therefore, the processing of the single-user tasks can be performed efficiently via our proposed processing strategy.

Figure 5.10 demonstrated the importance of integrating SemCom into SEC networks in future communication environments. I set the user and the TST to maintain the same status to transmit to LEO satellites in different rainfall environments. It can be observed that as the rainfall probability increases, the task transmission cost of TST without SemCom is exhibiting a significant increase. Because the Ka-band frequency is extremely high and is strongly influenced by rainfall-induced path loss. In contrast, the processing costs for users transmitting via C-band are only slightly increasing. Since the C-band frequency is smaller than the Ka-band frequency and thus tolerates less path loss. Nevertheless, the TST configuration with the semantic encoder spends the least processing cost. Furthermore, the processing cost did not increase significantly with the increase

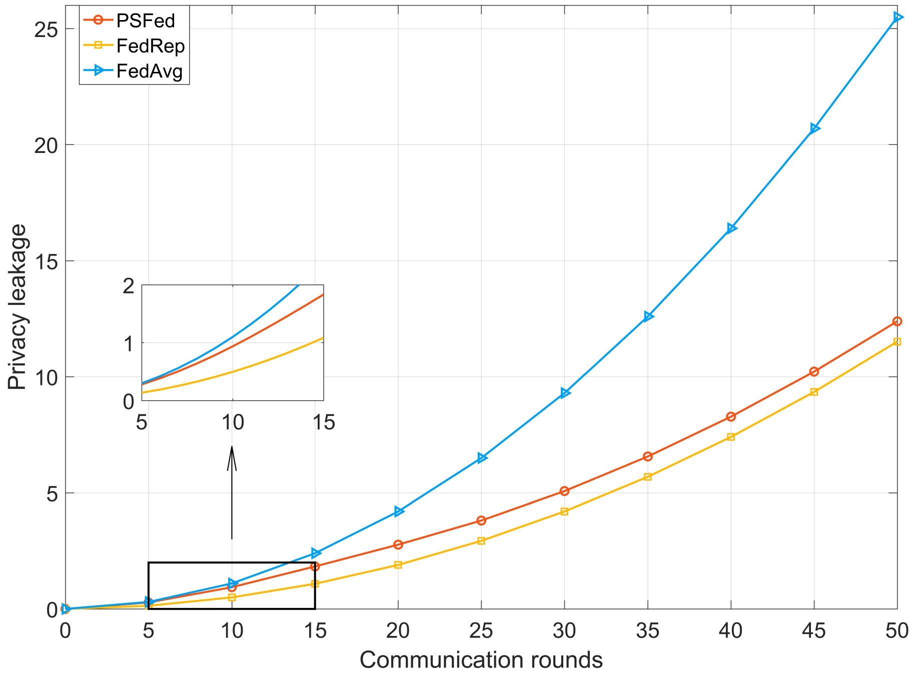


Figure 5.7: Privacy leakage of various learning approaches.

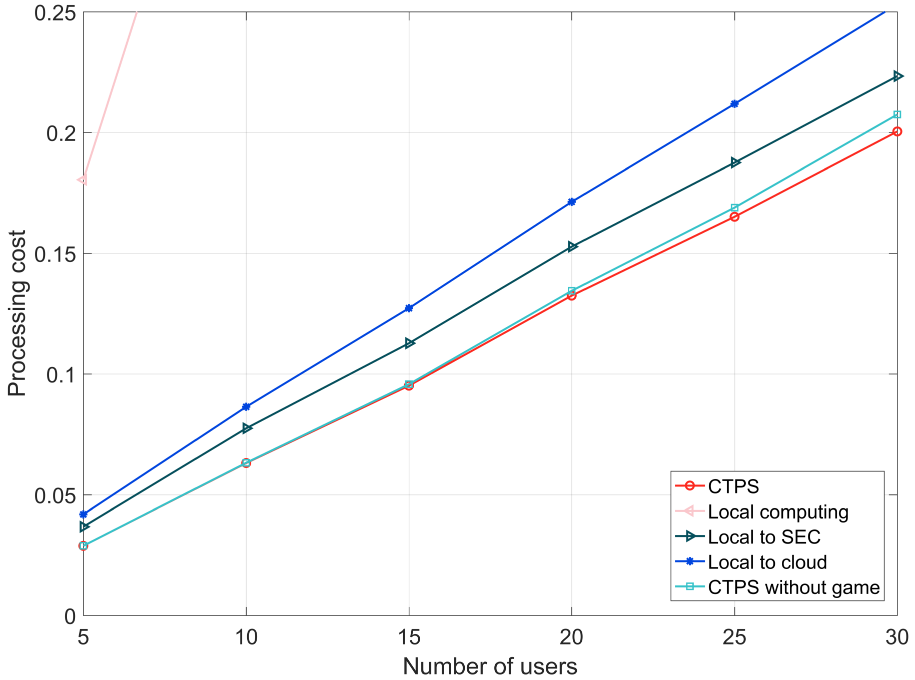


Figure 5.8: The processing cost of the varying number of users.

in rainfall rate. This is because the latency of semantic extraction is not affected by the environment. The improved spectrum efficiency also reduces the impact of rainfall-induced path loss. Therefore, the integration of SemCom in SEC networks is necessary.

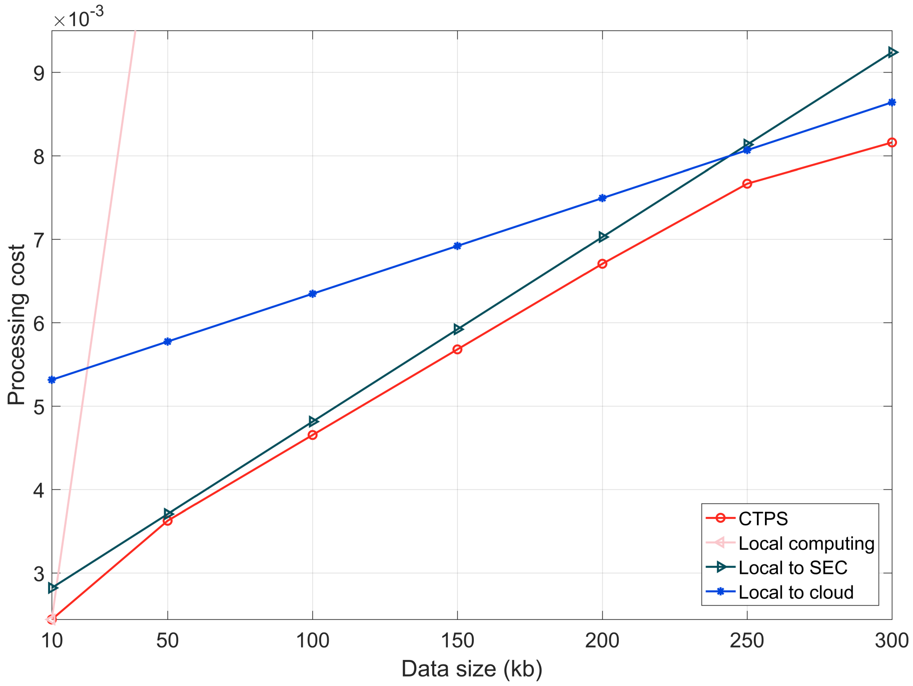


Figure 5.9: The processing cost of a single user.

In Figure 5.11, the influence of α and β on user strategies are investigated and the data size is from 5kb to 300kb simulated 50 times. The energy consumption weight β is always set as 0.5. I list the proportion of users that do not choose to offload via TST. It can be noticed that as the number of users increases, the unwillingness to offload increases due to the reduced number of subcarriers being allocated to them. However, users are always more reluctant to offload via TST in case the delay is more important (i.e., bigger α). These provide a criterion for the appropriate α and β to be chosen.

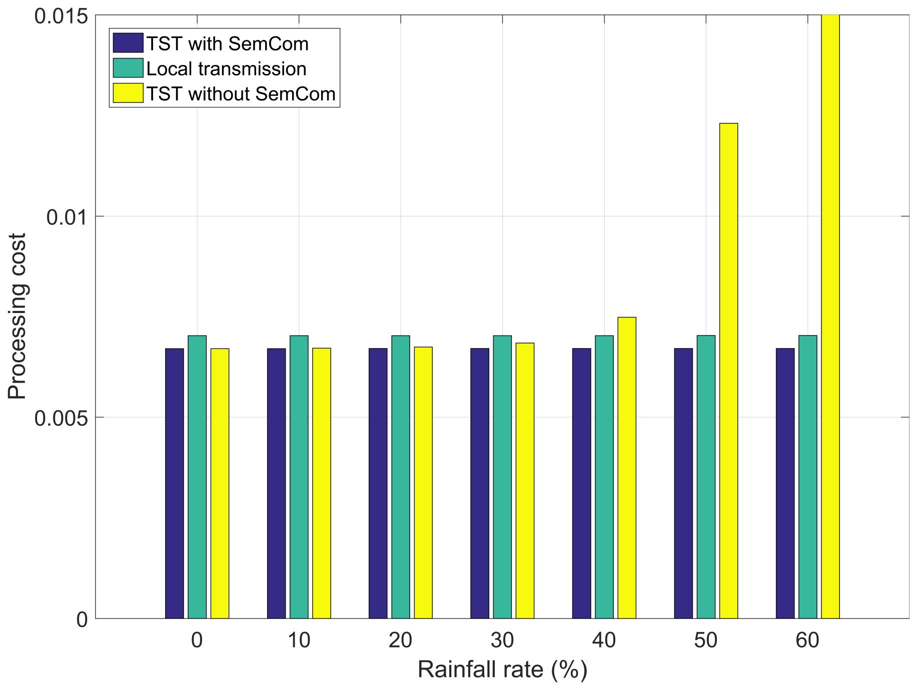


Figure 5.10: The usefulness of SemCom in the network.

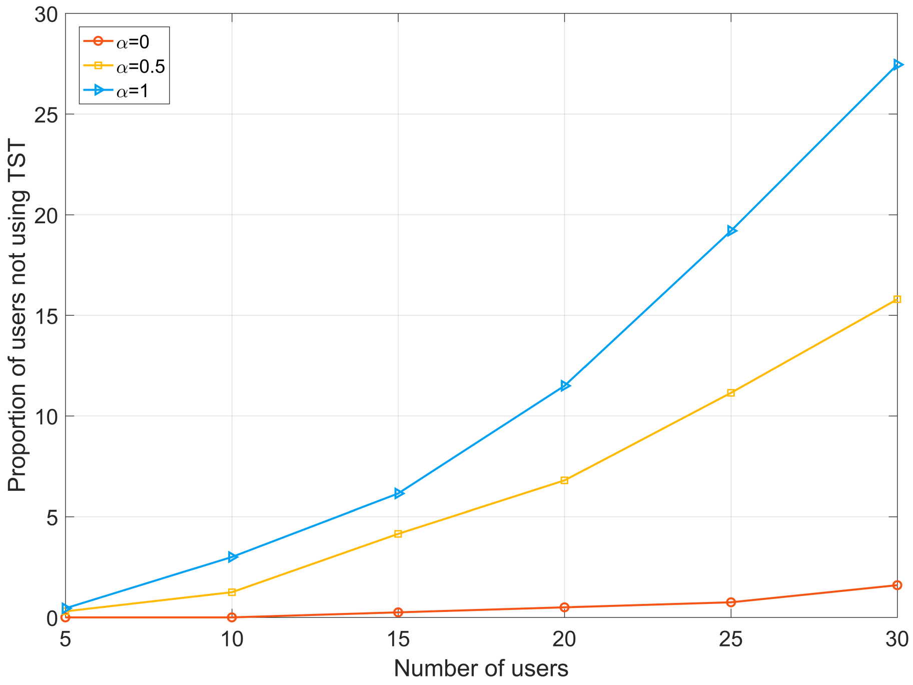


Figure 5.11: Impact of α and β on strategy developing.

5.6 Summaries

In this chapter, I investigated the integration of SemCom and SEC networks for terrestrial resource-limited users' computation offloading. I further proposed a novel SemCom-SEC framework for computation offloading. In addition, I examined the challenges that SemCom confronts in the proposed framework. For analysis, I then considered the challenges in two different scenarios. For the in-maintenance SemCom service, I proposed PSFed for the semantic coder update challenge. In the in-service SemCom service, I presented a game theoretical CTPS mechanism for task processing decision challenges of users. Compared with the general learning approach for semantic coder updating in SEC networks, simulations studies indicate that, on average, the proposed PSFed saves 40.50% of communication resources and further reduces privacy risk by 51.43%. Nevertheless, the training accuracy and convergence speed of PSFed and the general learning approach almost remain the same.

Chapter 6

Conclusions and Future Work

6.1 Summary

This thesis mainly focuses on SemCom system design for several potential 6G 3D wireless networks, i.e., terrestrial vehicular networks, air-terrestrial networks, and SAT networks. The challenges of SemCom coder updating and updating resource allocation in VN were investigated first. The SemCom coder updating and resource consumption in AEN and SEC networks were then studied.

In more detail, the main contribution of this thesis can be summarised as follows:

In Chapter 3, the terrestrial vehicular SemCom system was investigated. A novel MSFTL framework was proposed based on vehicle task offloading scenarios in this chapter. To enable adaptation to the complex vehicle semantic communication, the proposed framework divides the training of the model into four parts and uses the proposed split-federated learning. To further improve training efficiency, model accuracy, and the ability to adapt in highly mobile environments, a new learning approach integrated into the proposed framework based on TL was also presented. Finally, to incorporate vehicle mobility and training delays, a high-mobility training energy optimisation mechanism based on a Stackelberg game was designed for MSFTL. The performance of the proposed schemes was also investigated through extensive simulations. The results validated the proposed approach and indicate its superiority compared to the conventional learning frameworks for semantic communication in vehicular networks.

In Chapter 4, an air-terrestrial SemCom system for AENs is proposed. The SemCom energy consumption in such AEN was first investigated mathematically. An EGTIM was proposed for improving the energy efficiency of the AEN for SemCom. In addition, for semantic coders updating accurately and efficiently in the AEN with Non-IID training data, a GEDL framework was presented based on the renewed EGTIM. The simulation results confirmed the effectiveness of our proposed EGTIM in improving energy efficiency. In addition, the presented GEDL achieved outstanding performance in terms of increasing model training accuracy with Non-IID training data and decreasing training energy consumption.

In Chapter 5, a novel SemCom-SEC framework was proposed for the computation offloading of resource-limited users in SAT networks. An adaptive PSFed method for updating the semantic coder in SemCom-SEC was then proposed. The proposed method guarantees training convergence speed and accuracy. This method also improves the privacy of the semantic coder while reducing training delay and energy consumption. In the case of trained semantic coders in service, for the users processing computational tasks, the main objective is to minimise the users' delay and energy consumption, subject to sustaining users' privacy and fairness amongst them. This problem was then formulated as an incomplete information MINLP. A new CTPS mechanism was also proposed based on the Rubinstein bargaining game. Simulation results demonstrated the proposed PSFed and game theoretical CTPS mechanism outperforms the baseline solutions reducing delay and energy consumption while enhancing users' privacy.

6.2 Future work

Based on the current outcome of this thesis, the existing work and some promising topic directions can be further expanded in future works, summarised as follows:

1. *Secure SemCom*: Secure communication is one of the necessary requirements for 6G communication. The goal-oriented SemCom coders are required to conduct cooperative updates in the network in real-time, which raises numerous security concerns. In Chapter 4 and Chapter 5, the privacy of training data and SemCom coder were considered. Nevertheless, it is noteworthy that the proposed measures for privacy considerations are based on reducing the proportion of information exposed and thus reducing the risk of privacy leakage. It does not guarantee the security of SemCom. The security schemes to protect the training and transmission of SemCom transmission data and coders are thus urgently needed.

2. *Long-term resource allocation*: Emerging SemCom technologies change the resource consumption balance of traditional communication. Communication and computation resources need to be rebalanced, as discussed in Chapter 3 - Chapter 5. However, the provided mechanisms based on game theories for coder updating resource allocation consider short-term optimal strategies. Long-term optimal resource allocation strategies for SemCom also deserve elaborate studies via constructing the Markov decision process and utilising DRL.

3. *Life-long learning for SemCom*: Different SemCom coders are required to be updated for different specific transmission content. Adding new semantic transmission content means adding a new coder model that needs to be stored. This creates a continuously increasing storage load. In Chapter 2, although the TL approach was integrated into the proposed framework, a new storage model still be generated. How to continuously update new SemCom coder content on an identical SemCom coder without increasing the storage load is still an open challenge.

References

[1] E. C. Strinati, S. Barbarossa, T. Choi, *et al.*, “6g in the sky: On-demand intelligence at the edge of 3d networks,” *arXiv preprint arXiv:2010.09463*, 2020.

[2] I. F. Akyildiz, A. Kak, and S. Nie, “6g and beyond: The future of wireless communications systems,” *IEEE access*, vol. 8, pp. 133 995–134 030, 2020.

[3] S. S. Shinde, A. Bozorgchenani, D. Tarchi, and Q. Ni, “On the design of federated learning in latency and energy constrained computation offloading operations in vehicular edge computing systems,” *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 2041–2057, 2021.

[4] Y. Xiao, G. Shi, Y. Li, W. Saad, and H. V. Poor, “Toward self-learning edge intelligence in 6g,” *IEEE Communications Magazine*, vol. 58, no. 12, pp. 34–40, 2020.

[5] S. R. Pokhrel and J. Choi, “Understand-before-talk (ubt): A semantic communication approach to 6g networks,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 3, pp. 3544–3556, 2023.

[6] C. E. Shannon, “A mathematical theory of communication,” *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[7] Z. Qin, X. Tao, J. Lu, W. Tong, and G. Y. Li, “Semantic communications: Principles and challenges,” *arXiv preprint arXiv:2201.01389*, 2021.

[8] Q. Lan, D. Wen, Z. Zhang, *et al.*, "What is semantic communication? a view on conveying meaning in the era of machine intelligence," *Journal of Communications and Information Networks*, vol. 6, no. 4, pp. 336–371, 2021.

[9] J. Liang, Y. Xiao, Y. Li, G. Shi, and M. Bennis, "Life-long learning for reasoning-based semantic communication," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2022, pp. 271–276.

[10] H. Xie and Z. Qin, "A lite distributed semantic communication system for internet of things," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 142–153, 2020.

[11] G. Shi, Y. Xiao, Y. Li, and X. Xie, "From semantic communication to semantic-aware networking: Model, architecture, and open problems," *IEEE Communications Magazine*, vol. 59, no. 8, pp. 44–50, 2021.

[12] Z. Qin, G. Y. Li, and H. Ye, "Federated learning and wireless communications," *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134–140, 2021.

[13] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[14] C. Chaccour, W. Saad, M. Debbah, Z. Han, and H. V. Poor, "Less data, more knowledge: Building next generation semantic communication networks," *arXiv preprint arXiv:2211.14343*, 2022.

[15] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[16] O. Y. Bursalioglu, G. Caire, and D. Divsalar, "Joint source-channel coding for deep-space image transmission using rateless codes," *IEEE Transactions on Communications*, vol. 61, no. 8, pp. 3448–3461, 2013.

[17] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[18] J. Shao and J. Zhang, "Bottlenet++: An end-to-end approach for feature compression in device-edge co-inference systems," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, IEEE, 2020, pp. 1–6.

[19] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2018, pp. 2326–2330.

[20] H. Xie, Z. Qin, G. Y. Li, and B.-H. Juang, "Deep learning enabled semantic communication systems," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2663–2675, 2021.

[21] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, "Joint source-channel coding for video communications," *Handbook of Image and Video Processing*, pp. 1065–1082, 2005.

[22] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, "Wireless semantic communications for video conferencing," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 230–244, 2022.

[23] Z. Weng and Z. Qin, "Semantic communication systems for speech transmission," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2434–2444, 2021.

[24] H. Xie, Z. Qin, and G. Y. Li, "Task-oriented multi-user semantic communications for vqa," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 553–557, 2021.

[25] P. Wang, J. Zhang, X. Zhang, Z. Yan, B. G. Evans, and W. Wang, "Convergence of satellite and terrestrial networks: A comprehensive survey," *IEEE Access*, vol. 8, pp. 5550–5588, 2020.

[26] P. Rahimi, C. Chrysostomou, H. Pervaiz, V. Vassiliou, and Q. Ni, "Joint radio resource allocation and beamforming optimization for industrial internet of things in software-defined networking-based virtual fog-radio access network 5g-and-beyond wireless environments," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 4198–4209, 2021.

[27] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving qos of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE network*, vol. 33, no. 1, pp. 70–76, 2019.

[28] E. C. Strinati, S. Barbarossa, T. Choi, *et al.*, "6g in the sky: On-demand intelligence at the edge of 3d networks," *arXiv preprint arXiv:2010.09463*, 2020.

[29] L. Yang, X. Chen, S. M. Perlaza, and J. Zhang, "Special issue on artificial-intelligence-powered edge computing for internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9224–9226, 2020.

[30] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, vol. 2, p. 2, 2016.

[31] X. Kang, B. Song, J. Guo, Z. Qin, and F. R. Yu, "Task-oriented image transmission for scene classification in unmanned aerial systems," *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5181–5192, 2022.

[32] W. J. Yun, B. Lim, S. Jung, *et al.*, "Attention-based reinforcement learning for real-time uav semantic communication," in *2021 17th International Symposium on Wireless Communication Systems (ISWCS)*, IEEE, 2021, pp. 1–6.

[33] J. Kang, H. Du, Z. Li, *et al.*, "Personalized saliency in task-oriented semantic communications: Image transmission and performance analysis," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 1, pp. 186–201, 2022.

[34] Y. Wang, J. Yang, X. Guo, and Z. Qu, “A game-theoretic approach to computation offloading in satellite edge computing,” *IEEE Access*, vol. 8, pp. 12 510–12 520, 2019.

[35] Z. Zhang, W. Zhang, and F.-H. Tseng, “Satellite mobile edge computing: Improving qos of high-speed satellite-terrestrial networks using edge computing techniques,” *IEEE network*, vol. 33, no. 1, pp. 70–76, 2019.

[36] B. Di, L. Song, Y. Li, and H. V. Poor, “Ultra-dense leo: Integration of satellite access networks into 5g and beyond,” *IEEE Wireless Communications*, vol. 26, no. 2, pp. 62–69, 2019.

[37] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, “A computation offloading strategy in satellite terrestrial networks with double edge computing,” in *2018 IEEE international conference on communication systems (ICCS)*, IEEE, 2018, pp. 450–455.

[38] Z. Song, Y. Hao, Y. Liu, and X. Sun, “Energy-efficient multiaccess edge computing for terrestrial-satellite internet of things,” *IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14 202–14 218, 2021.

[39] Q. Tang, Z. Fei, B. Li, and Z. Han, “Computation offloading in leo satellite networks with hybrid cloud and edge computing,” *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164–9176, 2021.

[40] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, “Resource allocation for text semantic communications,” *IEEE Wireless Communications Letters*, vol. 11, no. 7, pp. 1394–1398, 2022.

[41] C. Liu, C. Guo, Y. Yang, and N. Jiang, “Adaptable semantic compression and resource allocation for task-oriented communications,” *arXiv preprint arXiv:2204.08910*, 2022.

[42] L. Yan, Z. Qin, R. Zhang, Y. Li, and G. Y. Li, “Qoe-aware resource allocation for semantic communication networks,” in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, IEEE, 2022, pp. 3272–3277.

[43] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, “Split learning for health: Distributed deep learning without sharing raw patient data,” *arXiv preprint arXiv:1812.00564*, 2018.

[44] M. J. Khabbaz, W. F. Fawaz, and C. M. Assi, “A simple free-flow traffic model for vehicular intermittently connected networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1312–1326, 2012.

[45] A. P. Miettinen and J. K. Nurminen, “Energy efficiency of mobile clients in cloud computing,” in *2nd USENIX workshop on hot topics in cloud computing (HotCloud 10)*, 2010.

[46] L. Xia, Y. Sun, X. Li, G. Feng, and M. A. Imran, “Wireless resource management in intelligent semantic communication networks,” in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2022, pp. 1–6.

[47] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention,” in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, 2016, pp. 223–228.

[48] T. O’shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[49] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *arXiv preprint arXiv:1610.05492*, 2016.

[50] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial intelligence and statistics*, PMLR, 2017, pp. 1273–1282.

[51] Z. Yu, J. Hu, G. Min, Z. Zhao, W. Miao, and M. S. Hossain, "Mobility-aware proactive edge caching for connected vehicles using federated learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5341–5351, 2020.

[52] D. Fudenberg and J. Tirole, *Game theory*. MIT press, 1991.

[53] P. J.-J. Herings and A. van den Elzen, "Computation of the nash equilibrium selected by the tracing procedure in n-person games," *Games and Economic Behavior*, vol. 38, no. 1, pp. 89–117, 2002.

[54] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[56] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-iid data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, 2021, ISSN: 0925-2312.

[57] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1397–1411, 2017.

[58] Q. Shi, L. Zhao, Y. Zhang, G. Zheng, F. R. Yu, and H.-H. Chen, "Energy-efficiency versus delay tradeoff in wireless networks virtualization," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 837–841, 2018.

[59] Q. Ni and C. C. Zarakovitis, "Nash bargaining game theoretic scheduling for joint channel and power allocation in cognitive radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 70–81, 2012.

[60] Y. Wang, W. Chen, T. H. Luan, *et al.*, "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1525–1539, 2022.

[61] H. Zhou, Z. Wang, G. Min, and H. Zhang, "Uav-aided computation offloading in mobile-edge computing networks: A stackelberg game approach," *IEEE Internet of Things Journal*, vol. 10, no. 8, pp. 6622–6633, 2023.

[62] R. Xing, Z. Su, and Y. Wang, "Intrusion detection in autonomous vehicular networks: A trust assessment and q-learning approach," *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 79–83, 2019.

[63] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017. arXiv: `1609.04747`.

[64] H. Yang, J. Zhao, Z. Xiong, K.-Y. Lam, S. Sun, and L. Xiao, *Privacy-preserving federated learning for uav-enabled networks: Learning-based joint scheduling and resource management*, 2020. arXiv: `2011.14197`.

[65] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, *Revisiting unreasonable effectiveness of data in deep learning era*, 2017. arXiv: `1707.02968`.

[66] L. D. Earley, "Communication in challenging environments: Application of leo/meo satellite constellation to emerging aviation networks," in *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, 2021, pp. 1–8.

[67] N. Zhang, S. Zhang, P. Yang, O. Alhussein, W. Zhuang, and X. S. Shen, "Software defined space-air-ground integrated vehicular networks: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 101–109, 2017.

[68] K. Tekbıyık, G. K. Kurt, and H. Yanikomeroglu, "Energy-efficient ris-assisted satellites for iot networks," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 891–14 899, 2022.

[69] F. Wang, J. Xu, and Z. Ding, "Multi-antenna noma for computation offloading in multiuser mobile edge computing systems," *IEEE Transactions on Com-*

munications, vol. 67, no. 3, pp. 2450–2463, 2019. DOI: 10.1109/TCOMM.2018.2881725.

[70] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, “Noma-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 244–12 258, 2018.

[71] S. Fu, J. Gao, and L. Zhao, “Integrated resource management for terrestrial-satellite systems,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3256–3266, 2020.

[72] ITU-R, “Propagation data and prediction methods required for the design of earth-space telecommunication systems,” *International Telecommunication Union (ITU)*, pp. 618–13, 2017.

[73] ITU-R, “Specific attenuation model for rain for use in prediction methods,” *International Telecommunication Union (ITU)*, pp. 838–3, 2005.

[74] F. Zenke, B. Poole, and S. Ganguli, *Continual learning through synaptic intelligence*, 2017. arXiv: 1703.04200.

[75] X. Ma, J. Zhang, S. Guo, and W. Xu, *Layer-wised model aggregation for personalized federated learning*, 2022. arXiv: 2205.03993.

[76] A. Rubinstein, “Perfect equilibrium in a bargaining model,” *Econometrica: Journal of the Econometric Society*, pp. 97–109, 1982.

[77] R. Deng, B. Di, and L. Song, “Pricing mechanism design for data offloading in ultra-dense leo-based satellite-terrestrial networks,” in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[78] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[79] Y. Pan, M. Chen, Z. Yang, N. Huang, and M. Shikh-Bahaei, “Energy-efficient noma-based mobile edge computing offloading,” *IEEE Communications Letters*, vol. 23, no. 2, pp. 310–313, 2019.

[80] K. Maine, C. Devieux, and P. Swan, “Overview of iridium satellite network,” in *Proceedings of WESCON’95*, 1995, pp. 483–.

[81] 3GPP, “Study on new radio (nr) to support non terrestrial networks (release 15),” in *Tech. Rep.*, 2017.

[82] Y. LeCun, C. Cortes, and C. Burges J.C., “The mnist database of handwritten digits,” 1998. [Online]. Available: `http://yann.lecun.com/exdb/mnist/`.

[83] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, *Exploiting shared representations for personalized federated learning*, 2023. arXiv: `2102.07078`.