Mobility-aware Split-Federated with Transfer Learning for Vehicular Semantic Communication Networks

Guhan Zheng, Qiang Ni, Keivan Navaie, Haris Pervaiz, Geyong Min,
Aryan Kaushik, and Charilaos Zarakovitis

***Abstract*—Machine learning-based semantic communication is a promising enabler for future-generation wireless network systems such as 6G networks. In practice, effective semantic communication requires online training for unknown content. In highly mobile vehicular networks, however, reliable, and efficient model training becomes significantly challenging. The existing distributed learning approaches are also unable to effectively operate in highly dynamic vehicular semantic communication networks. To address these challenges, we propose a novel mobility-aware split-federated with transfer learning (MSFTL) framework based on vehicle task offloading scenarios in this paper. 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 new split-federated learning. Furthermore, to improve training efficiency, model accuracy, and the ability to adapt in highly mobile environments, we also present a new transfer learning approach integrated into the proposed framework. Particularly, we propose a high-mobility training resource optimisation mechanism based on a Stackelberg game for MSFTL to further reduce training costs and adapt vehicle mobility scenarios. We also investigate the performance of the proposed schemes through extensive simulations. The results validate the proposed approach and indicate its superiority compared to the conventional learning frameworks for semantic communication in vehicular networks.**

***Index Terms*—Vehicle semantic communication networks, split federated learning, transfer learning, Stackelberg game, resource optimisation.**

I. Introduction

6G communication network systems will support numerous challenging applications including intelligent transport and vehicular networks [1]. A tremendous amount of data will be generated and transmitted over the vehicular networks, and thus requires the introduction of edge cloud (EC) facilities to provide additional computing and caching resources for the vehicles [2], [3]. This advancement empowers vehicles to access these resources by offloading tasks like object/image recognition and spatial computing to the edge in real-time through a communication link. As a result, there is a notable increase in the communication loads between vehicles and the edge. This results in higher required spectrum efficiency and affects the required reliability, and quality of service (QoS) for the vehicular network. Therefore, effectively addressing the challenge of enhancing communication efficiency, reliability, and QoS for the future vehicular network emerges as a central concern in the ongoing evolution of 6G-enabled vehicular networks. 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 vehicular networks [4]. Semantic communication is a new intelligent communication paradigm for 6G and is considered a promising solution to address these challenges [5].

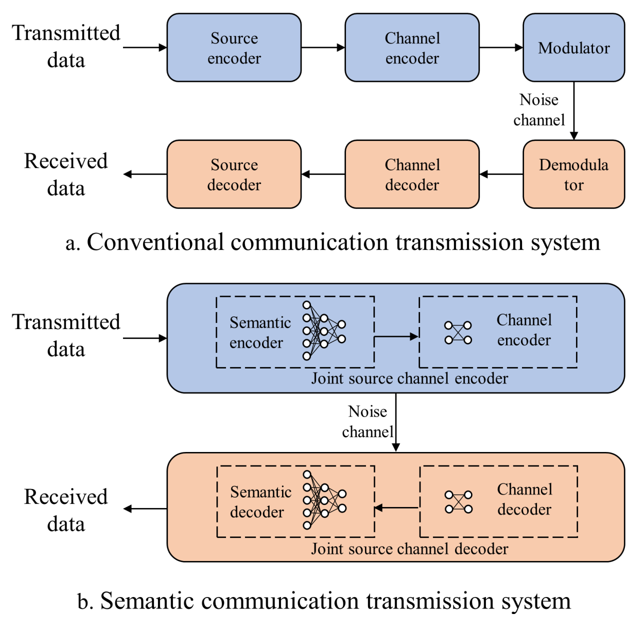

Fig. 1. Semantic versus conventional communication transmission systems.

Different from the conventional Shannon paradigm [6], semantic communication 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

Manuscript received 5 December 2023; revised 2 January 2024; accepted 17 January 2024. (Corresponding author: Qiang Ni.)

G. Zheng, Q. Ni and K. Navaie are with the School of Computing and Communications, Lancaster University, LA1 4WA, UK (Email:g.zheng2, q.ni, k.navaie@lancaster.ac.uk).

H. Pervaiz is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, UK (Email: haris.pervaiz@essex.ac.uk).

G. Min is with the Department of Computer Science, University of Exeter, EX4 4QF, UK. (Email: g.min@exeter.ac.uk).

A. Kaushik is with the School of Engineering and Informatics, University of Sussex, BN1 9RH, UK (Email: aryan.kaushik@sussex.ac.uk).

C. Zarakovitis is with the National Centre for Scientific Research Demokritos, Greece. (Email: c.zarakovitis@iit.demokritos.gr).

(DL) approaches. Using this approach, the network spectral efficiency is significantly reduced and transmission message reliability is dramatically increased [7]. It thus can improve the performance of the vehicular communication network and is one way to address these issues efficiently. Exploring the integration of semantic communication into vehicular networks is essential.

The existing research works are generally focused on designing semantic communication systems as a DL-based joint source-channel (DLJSC) coder to substitute the conventional transmission system [8] (Fig. 1). In this approach, the DLJSC encoder and decoder are deployed separately but are required to be trained for particular transmission contents together. Hence, the DLJSC transmission input and transmission output can be ensured to be the same. To do this, various semantic communication studies have been developed for image transmission [9], [10], text transmission [11], [12], video transmission [13], [14], speech [15], and visual question answering transmission [16]. These efforts demonstrated the excellent performance of the semantic communication systems in upgrading communication efficiency and transmission accuracy. Therefore, DLJSC encoders deployed on the vehicles and DLJSC decoders deployed on the ECs look promising to improve communication efficiency during task offloading of vehicles. Nevertheless, the existing proposed schemes are mostly point-to-point systems but fail to consider the application of semantic communication systems in actual multipoint networks. In practice, the DLJSC coder model needs to continue learning and updating on previously untrained content (e.g., image, speech and video), i.e., new content in the knowledge base (KB), to ensure providing a consistent QoS [17]. Conventional centralised training, however, may raise users' privacy concerns. Furthermore, a larger number of models can also occupy the scarce storage resources on the vehicles.

How to efficiently update semantic coders in the network in real-time is among the main challenges for semantic communication. Nevertheless, the existing studies are extremely limited in addressing this challenge of semantic communication. [5] and [18] investigated the possibility of applying the federated learning (FL) framework for general semantic communication networks. The FL framework is a privacy-preserving technology. It allows multiple clients to share the weight parameters of the trained model for joint training with multiple training data in a privacy-preserving manner. However, the deployment of existing FL-based semantic communication frameworks [5], [18] in vehicular semantic communication 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.

Several other distributed learning frameworks may have the potential to replace the existing FL-based framework for semantic communication. Vepakomma et. al [19] presented split learning (SL). In SL, part of the model is trained on distributed users and another part is trained on distributed central nodes. However, it is not applicable to vehicular semantic networks. Because the training data of vehicles for EC loss value calculation is unavailable to ECs due to privacy considerations. [20], [21] and [22] also introduced different frameworks integrating FL and SL. They, however, face the same problems as SL and cannot be used in vehicular semantic communication networks directly.

To the best of our knowledge, there is no previous effective learning framework able to update the semantic coder model in a real-time mobile vehicular offloading environment while addressing *Q1-Q3*, simultaneously.

In this paper, we propose a new split-federated learning framework for vehicle semantic communication to address these *Q1-Q3* urgent needs. Moreover, to satisfy the high mobility of the vehicle, we propose a novel TL paradigm integrated into the presented framework to increase training efficiency, decrease training costs and enable the framework to adapt mobility scenarios. We refer to our proposed mobility-aware split-federated with transfer learning framework as MSFTL. In particular, a high-mobility training resource optimisation mechanism is also presented based on the Stackelberg game for MSFTL to further reduce training costs and adapt high-mobility scenarios. The main contributions of this paper are as follows:

- We propose a novel MSFTL framework for vehicular semantic communication networks. 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 unupdated semantic encoder model, the MSFTL increases the convergence speed and accuracy. It decreases the training computing and communication costs. This approach also reduces storage load and performs well on a few sample learning scenarios.
- A Stackelberg game-based resource optimisation mechanism is developed to further reduce the training cost and optimise the proposed framework. The most appropriate amount of training data is selected fairly for each vehicle and the entire network. It jointly considers factors such as vehicle residence time, computational load, and communication overhead.
- For verification of the effectiveness of the framework and optimisation mechanism, we employ a classic semantic communication model [9] as our training model. The task offloading scenario is set as object/image recognition task offloading. We compare the simulation results of

our method with the existing frameworks for semantic communications and demonstrate the outperforming of our approaches.

The rest of the paper is organised as follows: Section II presents the related works and Section III presents the vehicle system model. The proposed MSFTL framework and the analysis of its computing and communication overhead are presented in Section IV. In Section V, the game theoretical mechanism design is proposed for resource optimisation. Section VI presents the simulation results showing that our proposed framework and mechanism achieve excellent performance. Finally, the paper is concluded in Section VII.

II. Related Works

Due to the lack of a distributed learning framework for semantic coder updating over vehicular semantic communication networks, we first discuss the feasibility of existing frameworks for semantic coder updating in general networks. Xie and Qin [23] proposed a lite distributed semantic communication system under power and latency constraints. It demonstrated the possibility of applying semantic communication in the Internet of Things (IoT) environment. However, this framework is suitable for the semantic coder model already updated rather than the coder needs to update. Based on it, Qin et al. [18] further proposed employing the FL framework for updating semantic coders. It, however, was totally based on the FL and mainly discussed the possibility of FL in semantic coder updating. This framework thus faces the same *Q1, Q2 and Q3* challenge as FL. Shi et al. [5] also introduced an FL framework for semantic communication networks where clients' DLJSC encoding and decoding take place at the EC. However, in this method, the clients need to transmit the original signal to the EC which does not fundamentally solve the problem of inefficient communication. Therefore, the existing framework for semantic communication in general networks is not applicable in vehicular networks.

Several proposed distributed frameworks with potential applications to vehicular semantic communication networks as well. Vepakomma et. al [19] presented split learning (SL). Nevertheless, this method needs vehicles to upload the training data to the edge. It is unacceptable due to privacy considerations. Thapa et. al [20] proposed a split-federated learning framework. Compared with SL, the model on distributed central nodes added the FL aggregation step. Furthermore, Romanini et. al [21] introduced a federated learning framework with splitNN. It is similar to [20]. They, however, face the same problems as SL, i.e., privacy data leakage. Hong et.al [22] also proposed a split-federated learning framework for different training domains. It splits a model for different domains and aggregates split models after training it on the user's end. It's still essentially FL and encounters the same challenges, i.e., *Q1, Q2 and Q3*, if employed in vehicular semantic communication networks. Therefore, there is still the absence of an effective learning framework that can be applied to semantic communication updating in vehicular networks.

III. System Model

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

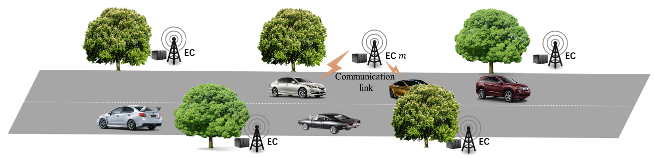

Fig. 2. Vehicles in the network.

A. Vehicle model

A set of ECs, $\{1, 2, ..., m, ..., M\}$, 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 EC m (Fig. 2). Further, there are I_m vehicles in EC 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 EC, where the EC receives it via a DLJSC decoder. When the vehicle or EC semantic knowledge base is scarce, vehicles need to be selected for participation in the training based on the vehicle's velocity. We assume the arrival of vehicles to each edge service range follows the widely used Poisson distribution and vehicles' average velocity is related to the degree of crowdedness on the road [24], [25]. We thus have the average velocity (km/h) $\bar{v}_m$ of N_m vehicles in the service range of EC m as:

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

where $v_{m_{max}}$ is the maximum vehicle velocity that can be driven within the service range of EC m. We assume roads in the EC 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 EC 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 EC m, $v_{n,m}$ is a normally distributed random variable with the probability density function given by [24]

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

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. We can also rewrite it as:

$$\begin{aligned} \hat{f}(v_{n,m}) &= \frac{f(v_{n,m})}{\int_{v_{m_{min}}}^{v_{m_{max}}} f(v_{n,m}) dv_{n,m}} \\ &= \frac{2f(v_{n,m})}{\mathrm{erf}(\frac{v_{m_{max}} - \bar{v}_m}{\sqrt{2}\sigma}) - \mathrm{erf}(\frac{v_{m_{min}} - \bar{v}_m}{\sqrt{2}\sigma})}. \end{aligned} \tag{3}$$

B. Computational and communication workload

We consider a vehicle computing offloading scenario, where vehicle n in the service range of EC m has a task with data size $k_{n,m}$ to offload. Further, we assume the size of training data to be computed by this vehicle during coder model training is $d_{n,m}$. We write the training delay of one epoch as:

$$T_{n,m} = \frac{d_{n,m}}{f_{n,m}}, \tag{4}$$

where $f_{n,m}$ is the CPU-cycle frequency of vehicle n with the unit cycles/s. The energy cost of computing is [26]

$$E_{n,m} = p_{n,m}^{c} T_{n,m} = \varepsilon f_{n,m}^{3} \frac{d_{n,m}}{f_{n,m}} = \varepsilon d_{n,m} f_{n,m}^{2}, \tag{5}$$

where ε is the energy parameter depending on chip [27] and $p_{n,m}^{c}$ is computing power.

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

$$t_{n,m} = \frac{k_{n,m}}{R_{n,m}} = \frac{k_{n,m}}{B_{n,m}\log_2(1+\frac{p_{n,m}g_{n,m}}{\sigma_0^2})}, \tag{6}$$

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

$$e_{n,m} = p_{n,m} t_{n,m}. \tag{7}$$

Semantic communication differs from traditional communication in spectral efficiency research [28], [29]. Conventional communications focus on unit bandwidth rates, while semantic communications focus on effective semantic information delivered per second. We also consider that in practical signal transmission, the transmission process of semantic communication is still based on traditional communication theory as described above.

For easy reference, the main parameters and their description used throughout this paper are presented in Table I.

TABLE I
NOTATION DEFINITION

Symbol	Definition
M	Set of ECs
N_m	Set of vehicles in the service range of EC m
$d_{n,m}$	Training data size of vehicle n
$f_{n,m}$	CPU-cycle frequency of vehicle n
$T_{n,m}$	Training delay of one epoch
$E_{n,m}$	Energy cost of computing
$k_{n,m}$	Transmission data size of vehicle n
$t_{n,m}$	Communication delay
$e_{n,m}$	Transmission energy cost
P_1	Pre-training model
P_2	Fine-tuning layers
P_3	EC private decoder
P_4	Last layer of the decoder
$\boldsymbol{x^{n,m}}$	Training samples of vehicle n
$\tilde{\boldsymbol{x}}_1^{\boldsymbol{n,m}}$	Pre-training model output of forward propagation
$\tilde{\boldsymbol{x}}_2^{\boldsymbol{n,m}}$	EC private decoder output of forward propagation
$\tilde{\boldsymbol{x}}_3^{\boldsymbol{n,m}}$	Decoder output of forward propagation
C^{FL}	Communication cost of FL
C^{MSFTL}	Communication cost of MSFTL
$D_{n,m}$	Maximum available training data from vehicle n
$\Psi_{n,m}$	Training duration of vehicle n
$K_{n,m}$	Vehicle residence time
$\Phi_{n,m}$	Energy cost of vehicle n during training
$\mu_{n,m}$	Utility function of the game at vehicle n
U_m	Utility function of the game at the EC m

IV. MSFTL FOR VEHICLE SEMANTIC COMMUNICATION

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

A. Transfer learning for vehicle semantic communication network

The successful application of Autoencoder, a deep unsupervised learning model, has recently been demonstrated in the design of semantic communication architectures [10], [30], [31]. 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. It fails to meet the demands of rapid updates to the highly mobile vehicular networks.

To address these challenges, we propose a TL approach. In this approach, we 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 cost for different missions is thus reduced.

B. MSFTL design

Considering the pervasive case of semantic coders update, we propose a novel training framework based on split-federated learning for vehicle semantic communication networks. As mentioned previously, existing SL-based frameworks are 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-based frameworks require identical models for federated aggregation which means FL require the same

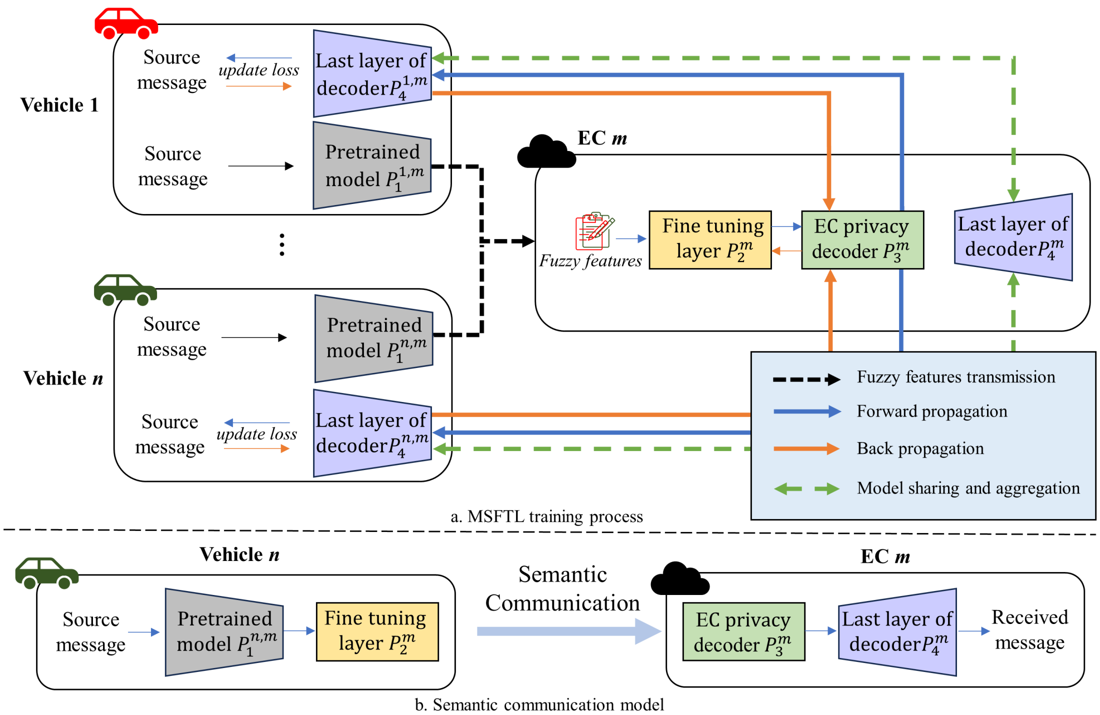

Fig. 3. The framework of the proposed MSFTL.

encoder model in our considered vehicular semantic networks. Therefore, based on the above, neither of these traditional frameworks can be applied to the vehicle semantic communication network as they face the *Q1-Q3* and privacy challenges. In our proposed MSFTL (Fig. 3), 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 EC private decoder (part of the decoder) P_3 and the last layer of the decoder P_4. The entire model is split but trained together.

The semantic communication model update algorithm is shown in Algorithm 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 $\tilde{\boldsymbol{x}}_{\boldsymbol{1}}^{\boldsymbol{n,m}}$ are first extracted from training samples $\boldsymbol{x}^{\boldsymbol{n,m}}$. The features $\tilde{\boldsymbol{x}}_{\boldsymbol{1}}^{\boldsymbol{n,m}}$ 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 $\tilde{\boldsymbol{x}}_{\boldsymbol{1}}^{\boldsymbol{n,m}}$ as inputs and start the training cycle. In one epoch, the EC uses $\tilde{\boldsymbol{x}}_{\boldsymbol{1}}^{\boldsymbol{n,m}}$ 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.

C. 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 semantic communication network due to the *Q1-Q3* and privacy challenges. However, to enable the employment of FL, we can assume that the vehicle encoder models are the same. To further validate the advantages of our MSFTL in the following, we compare the computational and communication load of the existing FL framework with the proposed MSFTL for the same encoder model. Furthermore, the training epochs of FL and MSFTL are assumed as the same for intuitive

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 ▷ for federated aggregation
  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 ▷ weight parameters of P_3^m and P_4^m in epoch a, respectively
    Get $W_{4,n}^a$ from vehicles
    Update $W_{4,n}^a$ ▷ federated aggregation
  end for

comparison, although simulations would show that our training converges faster.

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

$$T_{n,m}^{FL} = D_{n,m}\frac{d_P}{f_{n,m}}epoch, \tag{8}$$

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

$$E_{n,m}^{FL} = \epsilon D_{n,m} d_P f_{n,m}^2 epoch. \tag{9}$$

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

$$T_{n,m}^{MSFTL} = D_{n,m}(\frac{d_{P_4^{n,m}}}{f_{n,m}}epoch + \frac{d_{P_1^{n,m}}}{f_{n,m}}), \tag{10}$$

$$E_{n,m}^{MSFTL} = \epsilon D_{n,m}(d_{P_4^{n,m}} f_{n,m}^2 epoch + d_{P_1^{n,m}} f_{n,m}^2), \tag{11}$$

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,

$$d_P > d_{P_1^{n,m}} + d_{P_4^{n,m}}. \tag{12}$$

We can also write:

$$T_{n,m}^{FL} > T_{n,m}^{MSFTL}, \tag{13}$$

$$E_{n,m}^{FL} > E_{n,m}^{MSFTL}. \tag{14}$$

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

We 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 EC and return them after EC aggregation in each training epoch. FL therefore communication load of vehicle n is [32]

$$C_{n,m}^{FL} = 2\omega_p epoch, \tag{15}$$

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

$$C1_{n,m}^{MSFTL} = 2\omega_{p_4^{n,m}} epoch, \tag{16}$$

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 EC, the EC and client need to perform forward and backpropagation of the final layer of the decoder. The split training communication load is therefore

$$C2_{n,m}^{MSFTL} = O_1^{n,m} D_{n,m} + 2O_3^{n,m} D_{n,m} epoch, \tag{17}$$

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

$$\begin{aligned} C_{n,m}^{MSFTL} &= C1_{n,m}^{MSFTL} + C2_{n,m}^{MSFTL} \\ &= 2epoch(\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}) + O_1^{n,m} D_{n,m}. \end{aligned} \tag{18}$$

Since $epoch$ is usually a large number, we have $2epoch(\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}) >> O_1^{n,m} D_{n,m}$. Therefore, we 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 $\omega_{p_4^{n,m}} + O_3^{n,m} D_{n,m}$. We 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 even in the same training epoch setting. It demonstrates its viability in the real-world compared to FL.

V. Stackelberg Game Based Resource Optimisation Mechanism

In this section, we present a high-mobility training resource optimisation mechanism for the MSFTL. We found that an increase in the size of a vehicle's training dataset increases the energy consumption of the vehicle during training. Therefore, vehicles are not always willing to provide sufficient training data for training and aggregation. However, a decrease in the size of a certain vehicle's dataset also reduces the accuracy of the model when aggregated by edges. An incentive mechanism to motivate vehicles to provide as much training data as possible while taking into account vehicle mobility is thus essential. Considering the fairness, the mechanism is based on the Stackelberg game, which jointly takes into account vehicle mobility and minimises training costs. First, we present the game at vehicles in the mechanism and the selection of training vehicles considering mobility. We then introduce the design of the game at the EC and present mechanism optimisation formulation and its solution.

A. Game design at the vehicles

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

$$\begin{aligned}\Psi_{n,m} =& D_{n,m}\Big(\frac{d_{P_4^{n,m}}}{f_{n,m}}e + \frac{d_{P_1^{n,m}}}{f_{n,m}} + \frac{zO_1^{n,m} + 2zO_3^{n,m}e}{B_{n,m}\log_2(1+\frac{p_{n,m}g_{n,m}}{\sigma_0^2})}\Big) \\ &+ \textstyle\sum_n^{I_m} D_{n,m}\big(\frac{d_{P_{2,3}^m}+d_{P_4^m}}{f_m}\big)e,\end{aligned} \tag{19}$$

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 EC 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, we set

$$\Gamma_{n,m} = \frac{d_{P_4^{n,m}}}{f_{n,m}}e + \frac{d_{P_1^{n,m}}}{f_{n,m}} + \frac{zO_1^{n,m} + 2zO_3^{n,m}e}{B_{n,m}\log_2(1+\frac{p_{n,m}g_{n,m}}{\sigma_0^2})}, \tag{20}$$

and

$$\Psi_{n,m} = D_{n,m}\Gamma_{n,m} + \sum_n^{I_m} D_{n,m}\Big(\frac{d_{P_{2,3}^m}+d_{P_4^m}}{f_m}\Big)e. \tag{21}$$

The vehicle residence time can be estimated as:

$$K_{n,m} = \frac{h_{n,m}}{\bar{v}_m}, \tag{22}$$

where $h_{n,m}$ is the distance that the vehicle n travels out of the EC m's service range according to different road. Moreover, $\bar{v}_m$ is the vehicles' average velocity in EC m's service range mentioned in Section II. As $h_{n,m}$ is difficult to estimate, $h_{n,m}$ is considered the shortest distance at multiple forks in the road. Therefore, trainable vehicles should satisfy $\Psi_{n,m} \leq K_{n,m}$, we have

$$\Psi_{n,m}(D_{n,m}^{max}) < K_{n,m}. \tag{23}$$

Once suitable trainable vehicles have been identified, semantic coder model training can be initiated. For vehicles, participation in training results in energy consumption. As the most important concern for the vehicle, we consider energy consumption as the training cost $\Theta_{n,m}$, the same as in [26]. Training energy consumption is in turn only related to the number of training data. Thus we have

$$\Theta_{n,m} = E_{n,m}^{MSFTL} + D_{n,m}\frac{zp_{n,m}O_1^{n,m} + 2zp_{n,m}O_3^{n,m}e}{B_{n,m}\log_2(1+\frac{p_{n,m}g_{n,m}}{\sigma_0^2})}. \tag{24}$$

Nevertheless, the vehicle is not necessarily willing to participate in the training due to the different situations faced. For instance, vehicles already have a semantic encoder of the training goal. Sufficient data is one of the guarantees of model accuracy. We therefore designed a game to motivate vehicles fairly to participate in training. Vehicles providing more training data could reduce more money they spend on edge services, i.e., bonuses from ECs. The utility $\mu_{n,m}$ of vehicle n's participation in the training thus can be denoted by

$$\mu_{n,m} = \alpha R_{n,m} - \beta\Theta_{n,m}, \tag{25}$$

where α and β are monetary factors enable $\alpha R_{n,m} \leq 1$ and $\beta\Theta_{n,m} \leq 1$. Further, $R_{n,m}$ is the bonus vehicle n received. To ensure fair allocation of bonuses, we use a weight-sharing model commonly used in the game bonuses design. We write

$$R_{n,m} = \frac{\omega_{n,m}D_{n,m}}{\sum_n^{I_m}\omega_{n,m}D_{n,m}}R_m, \tag{26}$$

where R_m is the total bonus from the EC 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 $\omega_{n,m}$.

This allows the utility function to be a pure numerical function and the utility value is a unitless number. We can further define the vehicles' game problem as:

Problem 1:

$$\max_{D_{n,m}} \quad \alpha R_{n,m} - \beta\Theta_{n,m}, \tag{27a}$$

$$s.t. \quad D_{n,m}^{max} > D_{n,m} \geq D_{n,m}^{min}. \tag{27b}$$

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

B. Game design at the EC

In this subsection, we design the game at the EC and its utility function. We assume the accuracy of the model is related to the amount of training data. The objective of the EC is to minimise the bonus offered while satisfying the minimum QoS (accuracy) after training. Because receiving more accurate information improves income. Without loss of generality, the EC m's utility is defined as:

$$U_m \triangleq \gamma\Omega\Big(\sum_n^{I_m} D_{n,m}\Big) - \delta R_m, \tag{28}$$

where γ and δ are normalisation factors and Ω is a function related to the accuracy of the training model. 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 (Fig. 8). In addition, predictably, the accuracy is 0 with 0 data and bound by the maximum accuracy Ω^{max}. We believe that the trend shows a logarithmic function trend and is bound by Ω^{max}. We thus use a logarithmic function to model the Ω as:

$$\Omega(\sum_{n}^{I_m} D_{n,m}) \triangleq \ln(1+\theta\sum_{n}^{I_m} D_{n,m}), \tag{29}$$

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 EC thus can be written as:

Problem 2:

$$\max_{R_m} \quad \gamma\ln(1+\theta\sum_{n}^{I_m} D_{n,m}) - \delta R_m, \tag{30a}$$

$$s.t. \quad R_m > 0, \tag{30b}$$

$$\Omega^{min} < \ln(1+\theta\sum_{n}^{I_m} D_{n,m}) \leq \Omega^{max}. \tag{30c}$$

C. Optimal Solutions and Equilibrium Analysis

Nash Equilibrium Existence: Problem 1 (follower) and problem 2 (leader) form a Stackelberg game. We assume $D^*_{n,m}$ and R^*_m are the optimal solutions for Problem 1, and Problem 2, respectively. Thus, the game needs to satisfy the following equation to reach Nash Equilibrium (NE) point(s)

$$\mu(D^*_{n,m}, R^*_m) \geq \mu(D_{n,m}, R^*_m), \tag{31}$$

$$U(D^*_{n,m}, R^*_m) \geq U(D^*_{n,m}, R_m). \tag{32}$$

It is found from Problem 1 that the strategy set at vehicles is compact and convex, and the utility function is continuous and concave in $D_{n,m}$. Thus, according to the Debreu-Glicksberg-Fan theorem, a pure NE exists [33].

We then employ classic backward induction to find SE points. The optimal strategies for vehicles are obtained first, followed by the optimal strategy for the EC. 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 (26a) with respect to $D_{n,m}$, we have

$$\frac{\partial \mu_{n,m}}{\partial D_{n,m}} = \alpha\frac{\omega_n\sum_{j,j\neq n}^{I_m}\omega_{j,m}D_{j,m}}{(\sum_{n}^{I_m}\omega_{n,m}D_{n,m})^2}R_m - \frac{\beta\Theta_{n,m}}{D_{n,m}}. \tag{33}$$

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

$$U_m = \gamma ln(1+\theta\sum_{n}^{I_m} f_{n,m}(D^*_{n,m}, R_m)) - \delta R_m. \tag{34}$$

Algorithm 2 Stackelberg game-based resource optimisation mechanism

Set the maximum number of iterations K, and learning rate θ
Set initial positive numbers for R and D_i
while $k < K$
  $D_i(k) \longleftarrow \sqrt{\frac{\alpha R\sum_{j,j\neq n}^{I_m}\omega_j D_j}{\omega_i H_i}} - \frac{\sum_{j,j\neq n}^{I_m}\omega_j D_j}{\omega_i}$ ▷ optimal $D_i(k)$ without constraints
  $D^*_i(k) \longleftarrow$ constraints and D_i ▷ optimal $D_i(k)$
  $U(k) \longleftarrow R(k)$ and $D^*_i(k)$ ▷ based on Eq. (33)
  $R(k+1) = R(k) + \theta$
end while
while $k < K$
  Find the maximum $U(k)$ and corresponding $R(k)$ and $D^*_i(k)$
end while
return $R(k)$ and $D^*_i(k)$

Due to the high complexity and multiple constraints, sub-games NE cannot be derived in a closed form. Therefore, we solve the game in two segments through numerical search. In the first step, we employ the simplicial method [34] to achieve 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 (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 algorithm complexity can be thought of as $O(K^2)$ and thus can be solved in polynomial time in real-world applications. The solution algorithm is shown in Algorithm 2.

VI. Simulation Results

In this section, we evaluate the performance of the proposed MSFTL and optimisation mechanism. First, we investigate the efficiency of the proposed MSFTL framework in terms of convergence speed, and accuracy. Since SL-based frameworks [20]–[22] cannot be implemented in a vehicular environment. Because they have to share vehicles' private training data to the edge. It's unacceptable. we thus compare the proposed MSFTL framework with the existing FL-based framework for semantic communications [5], [18]. We also compare MSFTL with centralised learning (CL) for more comprehensive evaluation. Then, the advantage of the presented optimisation mechanism based on the Stackelberg game is assessed in a variety of different scenarios.

A. MSFTL

We first elaborate on the simulation settings in evaluating the performance of our proposed framework and ignore the communication noise when training. All simulation settings are the same as the previous semantic communication study, i.e., [9]. First, the adopted semantic communication model is

TABLE II
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
EC private decoder	transConv+ ReLU	10
	transConv+ ReLU	32
	transConv+ ReLU	64
Final layer of decoder	transConv+ Sigmoid	128

based on convolutional autoencoder (CAE) for image transmission as shown in Table II. Further, training and pre-training datasets employed are CIFAR 10 and CIFAR 100 [35] image data sets, respectively. They are both composed of a 50,000 32x32 colour 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. The batch size of the training is 64 and the learning rate is set as 0.1.

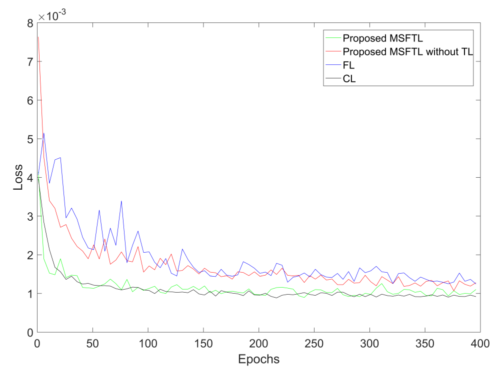

Fig. 4. Convergence speed comparison of different frameworks.

In order to more realistically verify the performance of the proposed framework in the case of vehicle task offloading, the experimental environment is set to object/image recognition after computing offloading. The validation of the transmitted images uses a fully trained VGG16 [36] network to classify, and its accuracy comparison with the images before transmission visualizes the performance of frameworks. We also assume the similarity of the recognition accuracy of the object/image after transmission in VGG16 compared to before transmission as the semantic communication model training 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. Since the baseline frameworks for semantic communication networks are limited and all based on FL, to enable the FL to operate in a vehicle semantic communication network, we ignore *Q1* for FL and assume all users have the same encoder model and the same degree of pre-training. In addition, we include CL as a comparison, in which the vehicle transmits all training data to the edge for centralised training. Even though it significantly compromises the privacy of the vehicle, it has the fastest convergence speed and highest accuracy and can be used as a distributed training benchmark. However, it also requires the same encoder model of vehicles. It is worth noting that our framework can be performed in any circumstance.

Fig. 4 illustrates the performance of the proposed MSFTL in terms of convergence speed. We set the batch size as 64 and compared the proposed MSFTL with the FL framework and the MSFTL without the TL model. To smooth out the results, we averaged the results for every 5 losses. The quicker stabilization of loss values indicates earlier convergence of training, enabling the prompt utilization of rapidly trained semantic coders in vehicular networks. We 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 and achieve almost similar convergence speed as CL. This is because the pre-training model accelerates the training and a well improves the model feature extraction capability.

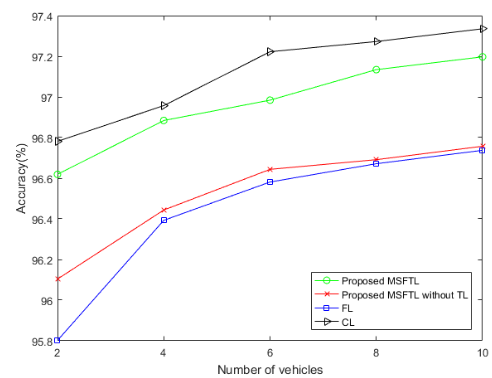

Fig. 5. Accuracy of different frameworks.

Fig. 5 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, as varying numbers of vehicles are involved in the training, our proposed MSFTL consistently achieves the optimal transmission/offloading accuracy that is second only to CL. The MSFTL achieving a high accuracy percentage not only signifies more accurate transmitted content but also highlights how trained semantic coders can enhance the accuracy of vehicle computing offloading, thereby improving the network quality of service. 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. (28) in our game design.

Fig. 6 shows the computing cost of the vehicle under different distributed training frameworks. For comparison purposes, we 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. Therefore, in our assessment, the cost (neuron number) has no unit but serves as an indicator of training delay and energy consumption. Lower costs imply that vehicles can complete training with reduced latency and energy consumption, enabling faster deployment of trained semantic coders. 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 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.

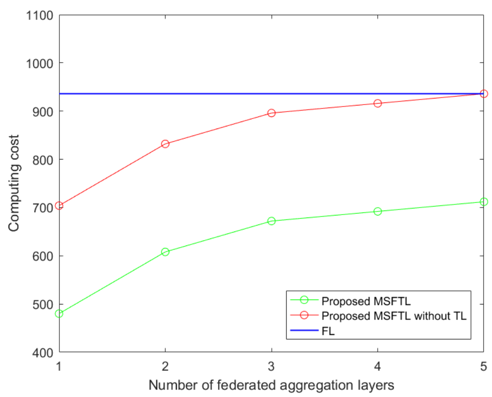

Fig. 6. Computing cost of different frameworks.

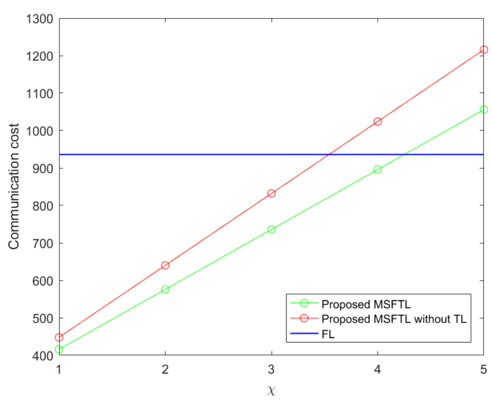

Fig. 7. Communication cost of different frameworks.

Fig. 7 evaluates the communication cost of the different distributed 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, we still assume that the federated aggregation communication cost is related to the number of neurons. The unit of cost is thus the same as computing cost. In addition, we 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. Similar to Fig. 6, the FL communication cost is independent of the amount of training data and thus remains fixed to a constant value. It can be seen that as χ increases, the communication cost of proposed MSFTL and MSFTL without TL also increases. Moreover, in case of 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.

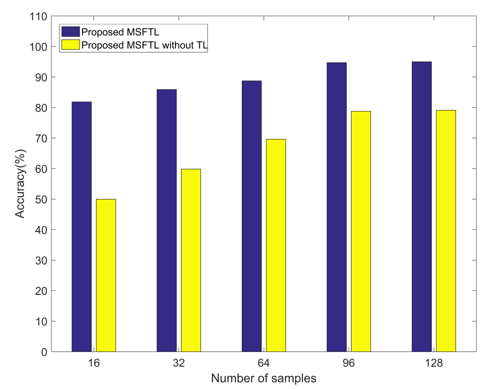

Fig. 8. Accuracy of different frameworks with sparse samples.

Fig. 8 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.

B. Game-based Resource Optimisation Mechanism

We 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, we assume all vehicles involved in the training have the same conditions (such as CPU cycles, velocity etc. Thus, in case Eq. (32) equals 0, Eq. (33) can be written as:

$$U_m = \gamma ln(1+\theta \frac{\alpha R_m(I-1)}{I_m H_{n,m}}) - \delta R_m. \tag{35}$$

We set $\gamma = 0.13$, $\alpha = 10$, $\delta = 0.08$ and $\theta = 8.5$ to approximate the simulation results in Fig. 5. 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, we 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 [37] and computational size required $d_{P_{2,3}^m} + d_{P_4^m}$ of EC m is 30 MB.

In Fig. 9, we investigate the influence of bonuses on the number of training data in different unit costs. We assume that the residence time of all vehicles is sufficient. It is seen that vehicles are less likely to participate in training at low bonus values. Because a 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 EC is limited by the maximum accuracy, so the amount of training data no longer changes.

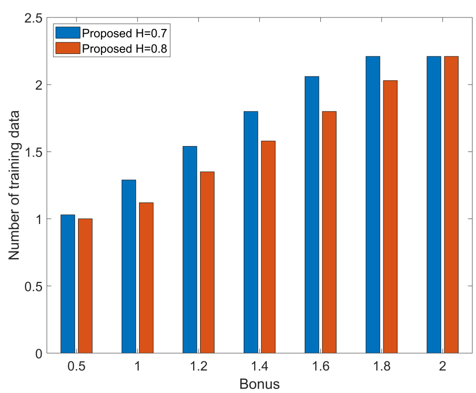

Fig. 9. Bonus impact on training data number.

Fig. 10 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 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.

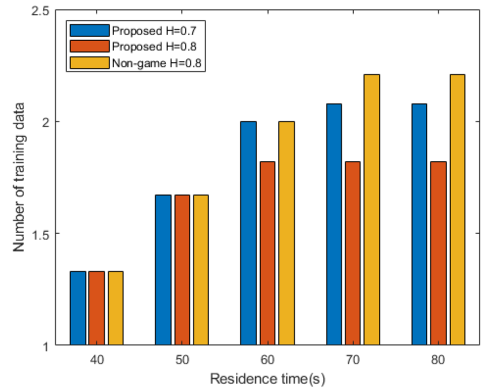

Fig. 10. Total training time versus various residence time.

VII. Conclusions

In this paper, we designed a new vehicle semantic communication framework, named MSFTL. It divides the trained DLJSC coder into four parts and utilises the proposed split federated learning for training. Different from existing frameworks for semantic communications, MSFTL can adapt to complex and various vehicle offloading scenarios. Further, in the proposed framework, we 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 costs. Moreover, an efficient high-mobility resource 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. We have also conducted simulation experiments to evaluate our proposed framework and resource optimisation mechanism. The simulation results demonstrated the effectiveness of our learning framework and mechanism. However, in the case of an extremely large number of training data sizes, MSFTL might confront heavy communication loads. We will try to address it in the future work.

References

[1] 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, Feb. 2022.

[2] L. Geng, H. Zhao, J. Wang, A. Kaushik, S. Yuan and W. Feng, ”Deep-Reinforcement-Learning-Based Distributed Computation Offloading in Vehicular Edge Computing Networks,” *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12416-12433, July 2023.

[3] Y. Liu, K. Xiong, Y. Lu, Q. Ni, P. Fan and K. B. Letaief, ”UAV-Aided Wireless Power Transfer and Data Collection in Rician Fading,” *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3097-3113, Oct. 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, December 2020.

[5] 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, August 2021.

[6] C. E. Shannon and W. Weaver, The Mathematical Theory of Communication. The University of Illinois Press, 1949.

[7] Z. Qin, X. Tao, J. Lu, and G. Y. Li, “Semantic communications: Principles and challenges,” Jun. 2022, [online] Available: http:// arXiv:2201.01389.

[8] Q. Lan 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, Dec. 2021.

[9] E. Bourtsoulatze, D. Burth 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, Sept. 2019.

[10] 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, August 2013.

[11] N. Farsad, M. Rao and A. Goldsmith, ”Deep Learning for Joint Source-Channel Coding of Text,” *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2326-2330.

[12] 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.

[13] F. Zhai, Y. Eisenberg, and A. K. Katsaggelos, “Joint source-channel coding for video communications,” *Handbook of Image and Video Processing*, 2nd ed., A. Bovik, Ed. Burlington, MA, USA: Academic, 2005.

[14] P. Jiang, C.-K. Wen, S. Jin, and G. Y. Li, “Wireless semantic communications for video conferencing,” Feb. 2022, [online] Available: http:// arXiv:2204.07790.

[15] 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, Aug. 2021.

[16] 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, March 2022.

[17] J. Liang, Y. Xiao, Y. Li, G. Shi, and M. Bennis, “Life-long learning for reasoning-based semantic communication,” Feb. 2022, [online] Available: http:// arXiv:2202.01952.

[18] Z. Qin, G. Y. Li and H. Ye, ”Federated Learning and Wireless Communications,” *IEEE Wireless Communications*, vol. 28, no. 5, pp. 134-140, October 2021.

[19] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, “Split learning for health: Distributed deep learning without sharing raw patient data,” 2018, [online] Available: http://arXiv:1812.00564

[20] C. Thapa, P. C. M. Arachchige, S. Camtepe, and L. Sun, “Splitfed: When federated learning meets split learning,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 8485–8493, 2022.

[21] D. Romanini, A. James Hall, P. Papadopoulos, T. Titcombe, A. Ismail, T. Cebere, et al., ”PyVertical: A vertical federated learning framework for multi-headed SplitNN”, arXiv:2104.00489, 2021.

[22] J. Hong, H. Wang, Z. Wang and J. Zhou, ”Efficient split-mix federated learning for on-demand and in-situ customization”, Proc. Int. Conf. Learn. Represent., pp. 1-18, 2021.

[23] 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, Jan. 2021.

[24] 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, Sept. 2012

[25] R. Xing, Z. Su, Q. Xu, N. Zhang and T. H. Luan, ”Secure Content Delivery for Connected and Autonomous Trucks: A Coalition Formation Game Approach,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 20522-20537, Nov. 2022.

[26] 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. 14202-14218, 15 Sept.15, 2021.

[27] A. P. Miettinen and J. K. Nurminen, ”Energy efficiency of mobile clients in cloud computing”, Proc. USENIX HotCloud, pp. 4-11, Jun. 2010.

[28] L. Yan, Z. Qin, R. Zhang, Y. Li and G. Y. Li, ”Resource Allocation for Text Semantic Communications,” *IEEE Wireless Communications Letters*.

[29] L. Xia, Y. Sun, X. Li, G. Feng, and M. A. Imran, “Wireless resource management in intelligent semantic communication networks,” Feb. 2022, [online] Available: http://arXiv:2202.07632.

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

[31] 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, Dec. 2017.

[32] A. Singh, P. Vepakomma, O. Gupta and R. Raskar, ”Detailed comparison of communication efficiency of split learning and federated learning”, *CoRR*, vol. abs/1909.09145, pp. 1-5, Sep. 2019.

[33] D. Fudenberg and J. Tirole, Game Theory. MIT Press, 1993.

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

[35] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. Int. Conf. on Learning Representations, San Diego, CA, 2015.

[37] 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, 1 June, 2021.