DRFSL: Deep Reinforced Federated Split Learning for Multi-Modal Beamforming in IoV

Jinxuan Chen*, Eric Samikwa*, Torsten Braun*, Kaushik Chowdhury†
*Institute of Computer Science, University of Bern, Switzerland
†Department of Electrical and Computer Engineering, University of Texas at Austin, USA
Email: *{jinxuan.chen, eric.samikwa, torsten.braun}@unibe.ch, † {kaushik}@utexas.edu

***Abstract*—In Vehicle-to-Everything (V2X) communication, advanced beamforming techniques address signal attenuation caused by mmWave, which provides high bandwidth and low latency. Multi-modal beamforming using Federated Learning (FL) can leverage resources like GPS, Lidar, and image data, significantly accelerating beam searching while enhancing data privacy. The heterogeneity of vehicles, however, affects the availability of computing resources for training machine learning models. Moreover, the multi-modal fusion network may contain billions of parameters, leading to extended training time for FL. To address these challenges, this paper proposes a novel Deep Reinforced Federated Split Learning framework (DRFSL) tailored for multi-modal beamforming with different sub-model architectures. DRFSL efficiently utilizes MEC computing and adapts the collaborative and distributed training to dynamic network conditions and system heterogeneity by incorporating deep reinforcement learning and split learning with FL. Experimental evaluation using real-world datasets demonstrates that DRFSL minimizes average training time by 49.45% and inference time by 24.43% and can achieve higher accuracy within the same timeframe compared to the existing FLASH framework.**

***Index Terms*—federated split learning, multi-modal beamforming, deep reinforcement learning, sector selection**

I. Introduction

Modern autonomous vehicles use multimodal perception technologies, such as GPS, LiDAR, and image sensors, to provide accurate environmental perception, positioning, and redundancy to optimize vehicle decision-making [1]. As they also rely on Vehicle-to-Everything (V2X) communication technology, the advent of 6G will provide higher data transmission rates, ultra-low latency, and large-scale connectivity, further enhancing the performance of autonomous driving [2]. Especially, Milimeter Waves (mmWave) play an important role in V2X communications and have the potential to provide high bandwidth and low latency for autonomous vehicles [3], meeting the expectations for applications like real-time high-definition map updates and high-definition video streaming.

However, due to the sparse multipath propagation in the mmWave band, it is more sensitive to being stuck into severe signal attenuation and weak penetration [4]. Advanced beamforming technology can overcome these drawbacks by precisely adjusting the beam direction and concentrating energy transmission, thereby enhancing signal coverage and communication reliability of autonomous vehicles in complex environments [5].

During beamforming, the antenna array adjusts the gain and phase of each element in real time. Recent studies show that using multi-modal sensor data can effectively predict the optimal sector or at least narrow down the search space, thereby accelerating beam training [6]–[8]. Nevertheless, these approaches do not incorporate real-world experiments with live sensor data. Additionally, the mentioned techniques rely on centralized systems, posing challenges such as high bandwidth requirements for data transfer over control channels, which are vulnerable to saturation and malicious interference.

Federated Learning (FL) [9] is an emerging distributed Machine Learning (ML) method where multiple vehicles can jointly train models without sharing original data, thus protecting data privacy and avoiding starving the control channel. Salehi et al. [10] proposed a multimodal FL framework, where local Deep Learning (DL) model weights are globally optimized by fusing them at the MEC. However, ML models encompass hundreds of millions of parameters trained on local vehicular systems. This complexity can impose a significant computational burden and result in prolonged training time, even during inference [11]. Furthermore, this issue is exacerbated by the heterogeneity of different vehicle systems, especially when vehicular resources are concurrently allocated to critical safety functions.

The application of Federated Split Learning (FSL) presents a versatile and flexible deployment strategy tailored to meet the unique demands of various devices [12]. By making full use of the Split Learning (SL), the training process is distributed among the devices and the servers, ensuring sensitive information stays on-premises while still leveraging collaborative model training through offloading techniques to optimize the computational resources [13]. To efficiently utilize FSL in multi-modal beamforming, it is essential to determine how to select the optimal split points of the different model branches, considering the heterogeneity of computing and communication resources on each vehicle and adapting respective split policies according to dynamic system conditions during training or inference.

In this paper, we propose Deep Reinforced Federated Split Learning (DRFSL) for mmWave multi-modal beamforming in Internet-of-Vehicle (IoV). DRFSL addresses key issues in multi-modal beamforming by introducing multi-modal SL into FL and integrating Deep Reinforcement Learning (DRL). Our framework utilizes multiple Multi-access Edge Computing -

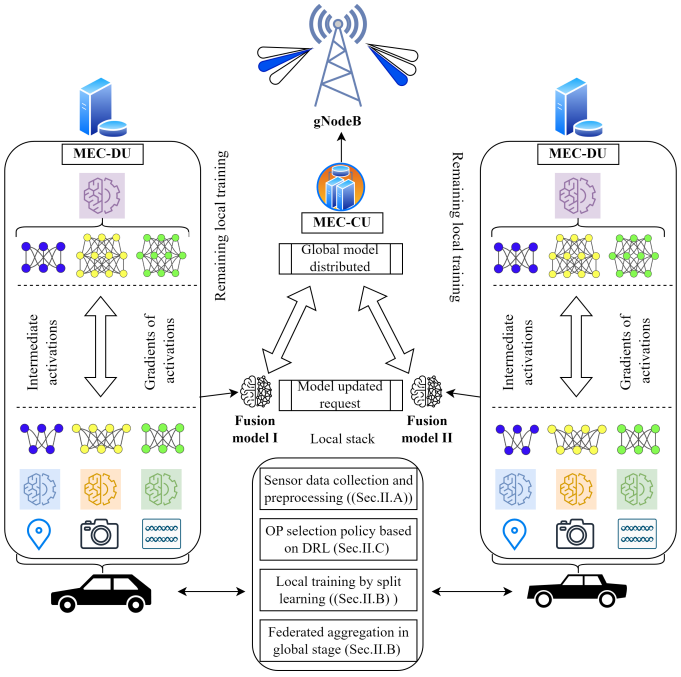

Fig. 1: The schematic of the proposed DRFSL framework integrating DRL with FSL assisted by MEC servers under 5G RAN for mmWave multi-modal beamforming in IoV.

Distributed Unit (MEC-DU) in 5G Radio Access Network (RAN) to enhance performance by distributing the training closer to end-users, thereby minimizing the time required for data transmission. MEC - Centralized Unit (MEC-CU) server is employed for central integration, ensuring efficient communication, as illustrated in Fig. 1. The split policies of the fusion models are determined by an optimization module assisted by DRL. The DRFSL framework effectively leverages multiple MEC-DU servers for offloading training and inference tasks and the MEC-CU server for global federated aggregation and distribution, all managed by gNodeB under 5G RAN.

To the best of our knowledge, there's no existing framework suitable for the multi-modal beamforming in the IoV scenario that can address all the problems, such as multi-fusion network splitting, handling strugglers due to the heterogeneous computing capability of the vehicles, and adapting to dynamic network conditions. Our main contributions are as follows:

- We propose DRFSL, specifically tailored for multi-modal beamforming scenarios. This approach leverages multiple MEC-DU servers for peer-to-peer communication with vehicles to prevent overloading, ensuring optimal performance for both vehicles and servers. Moreover, we leverage MEC-CU for updated weights, broadcasting, and distribution, reducing the latency during aggregation.
- We introduce multi-modal SL into FL training for the fusion network in IoV, where different sub-branches of the fusion architecture are efficiently split. It is assisted with a DRL optimization module to address system heterogeneity, mitigate stragglers, and adapt to dynamic networks. By intelligently distributing computational tasks, the DRFSL framework enhances system efficiency and reliability, effectively balancing the computational demands across the network.
- We evaluate DRFSL on real-world datasets and compare its performance to the existing multi-modal fusion network framework. The results demonstrate that DRFSL significantly reduces both training time and inference time under dynamic conditions in mobile scenarios by an average of 49.45% and an average of 24.43% improvement, respectively, compared to the FLASH framework.

The rest of the paper is organized as follows: Section II presents the system model, the formulation of the problems, and the DRL-based optimization module. Then, Section III provides a detailed explanation of the results obtained from real-world datasets. Finally, Section IV draws the conclusions.

II. Federated Split Learning for Multi-Modal Beamforming

A. System Model

The codebook is a predefined set of beamforming vectors that the transmitter and the receiver can use to direct signals towards specific directions, which can be defined as $C_{Tx} = \{t_1, ..., t_M\}$, and $C_{\text{Rx}} = \{r_1, ..., r_N\}$, where $M + N$ are probe frames of the mandatory Sector Level Sweep (SLS) during the IEEE 802.11ad standard sector initialization steps. Considering the downlink communication, through the best beam, we can obtain the best sector r^* with the maximum received signal strength, which can be derived as:

$$r^* = \underset{1 \le n \le N}{\arg\max}\, y_{r_n} \tag{1}$$

where y_{r_n} is the observed received signal strength at the Tx side when the receiver is configured as sector r_n. In modern 5G networks, multiple MEC servers near each gNodeB provide edge-computing and communication services [14], reducing latency by placing resources closer to users.

Consider a set of vehicles Φ in the coverage of the gNodeB, and they communicate with one of the MEC-DU servers. Each vehicle $\phi \in \Phi$ collects the dataset $D_\phi = \{X_{\phi,C}, X_{\phi,I}, X_{\phi,\zeta}\}$ through the multi-modal sensors GPS, Camera, and Lidar, respectively. We use the network architecture of the FLASH [10] whose fusion network $f^{\phi}_{\theta,FN}(\cdot)$ for each vehicle Φ parameterized by θ, containing for branches (GPS, Image, Lidar, Integrated): Γ_C, Γ_I, Γ_ζ, and Γ_ϖ. Each branch can be biased through the probability function $P^{(k)}$ by the selective strategy $\Gamma^{(k)}$ to provide a comprehensive situation-aware model and accurate performance.

The network architecture of the FL system is significantly enhanced through edge-device collaborative training, as illustrated in Fig. 2. By leveraging multiple MEC-DU servers, local models can be trained in parallel, greatly improving the overall efficiency of the network over FedAdapt [15], which relied on a single server for sequential training and failed for load balancing. MEC-DU servers periodically exchange and synchronize model parameters at the MEC-CU, which manages the MEC-DU servers to complete the global optimization model and distribute it to the vehicles.

B. Problem Formulation

1) Training Phase: During the distributed training stage, the number of training samples D_ϕ of the vehicle ϕ have been partitioned into equal-sized subsets minibatches ς, and we refer to j as the index of the iteration of local weight updates in the $k-th$ FL round. By using the mini-batch gradient descent, the weights of each round can be updated as follows:

$$w'_\phi = w_\phi - \eta \frac{1}{\varsigma} \sum_{i=1}^{\varsigma} \frac{\partial F(y^{(i)}, \hat{y}^{(i)})}{\partial w_\phi} \tag{2}$$

where η is the learning rate for which we use the StepLR [16] to adjust and optimize, $y^{(i)}$ represents the true labels, while $\hat{y}^{(i)}$ denotes the estimated labels, and F is the loss function. We define Θ as the time for computing the loss of the whole mini-batch.

On each vehicle ϕ, they share the same fusion DNN models of different branches of the sub-architecture, and we only consider the mixed split policies of the GPS, Image, and the Lidar models, composed of the N_C, N_I, N_ζ consecutive layers, respectively, denoted as follows:

$$\begin{aligned} &\{L_\mu^C | 1 \le \mu_{\phi,C}^{(k)} \le N_C, \mu \in \mathbb{N}\} \\ &\{L_\mu^I | 1 \le \mu_{\phi,I}^{(k)} \le N_I, \mu \in \mathbb{N}\} \\ &\{L_\mu^\zeta | 1 \le \mu_{\phi,\zeta}^{(k)} \le N_\zeta, \mu \in \mathbb{N}\} \end{aligned} \tag{3}$$

where e.g. $\mu_{\phi,C}^{(k)}$ is a split for the GPS branch model C. Note that not all the Offloading Points (OPs) are valid partitioning points, for example, if the DNN model has blocks, which means layers in the parallel path [17]. Therefore, we define N as the total number of valid OPs for each branch. We refer to $\mu_\phi^{(k)}$ as the *mixed split policy* for all model branches:

$$\mu_\phi^{(k)} = \{\mu_{\phi,C}^{(k)}, \mu_{\phi,I}^{(k)}, \mu_{\phi,\zeta}^{(k)}\} \tag{4}$$

To efficiently facilitate distributed learning between the vehicles and the MEC-DU, the fusion architecture is partitioned into two DNN sub-models. The first part of the model sub-branches is trained locally on the vehicle, consisting of layers $\{L_i^C | 1 \le i \le \mu_{\phi,C}^{(k)}, i \in \mathbb{N}\}$,$\{L_i^I | 1 \le i \le \mu_{\phi,I}^{(k)}, i \in \mathbb{N}\}$, and $\{L_i^\zeta | 1 \le i \le \mu_{\phi,\zeta}^{(k)}, i \in \mathbb{N}\}$. The time for forward propagation and backpropagation on this part is computed as $\delta_\phi^{(k)}$ and ${\delta'}_\phi^{(k)}$, respectively. The resulting intermediate activations V_μ also named smash data can be expressed as:

$$V_\mu = \sum_{i \in \psi} V_{\mu,i} \tag{5}$$

where $\Psi = \{C, I, \zeta\}$ is the index set of model branches. It is then sent with an average network throughput $B_{\phi,e}^{(k)}$ to a selected MEC-DU e, the unit that processes the layers $\{L_i^C | \mu_{\phi,C}^{(k)} + 1 \le i \le N_c, i \in \mathbb{N}\}$, $\{L_i^I | \mu_{\phi,I}^{(k)} + 1 \le i \le N_I, i \in \mathbb{N}\}$ and $\{L_i^\zeta | \mu_{\phi,\zeta}^{(k)} + 1 \le i \le N_\zeta, i \in \mathbb{N}\}$. The time for the continuous forward propagation and the backpropagation on the MEC-DU are defined as ${\delta_e}^{(k)}$ and ${\delta'}_e^{(k)}$, respectively.

Moreover, during each FSL round, the split policy is maintained unchanged while the vehicle is in motion, with an average throughput of $B_{e,\phi}^{(k)}$. The gradients of the smash data ${V'}_\mu$ are sent back from the MEC-DU servers to the vehicles. It is important to note that certain FL training rounds are completed when the moving vehicles remain within the coverage of the same gNodeB.

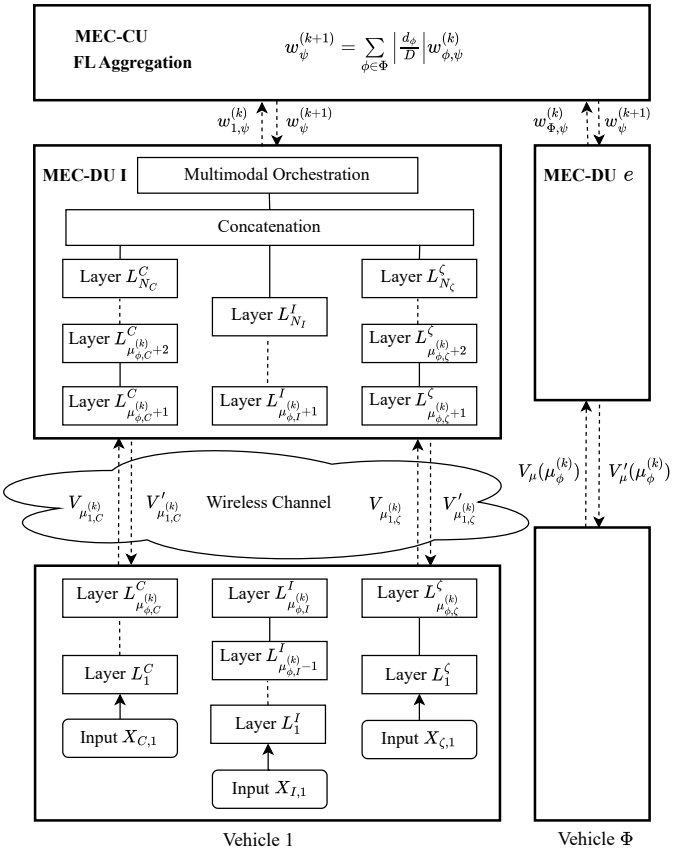

Fig. 2: Logical-level designed DRFSL framework for multi-modal beamforming in FL rounds with multiple MEC-DU servers.

Therefore, the time of local training of each vehicle in the FSL process $\Delta_\mu^{(k)}(\phi)$ can be expressed as:

$$\Delta_\mu^{(k)}(\phi) = \sum_{j=1}^{J} \frac{D_\phi}{\varsigma} (\delta_{\phi,e}^{(k)} + {\delta'}_{\phi,e}^{(k)} + \frac{V_\mu(\mu_\phi^{(k)})}{B_{\phi,e}^{(k)}} + \frac{V'_\mu(\mu_\phi^{(k)})}{B_{e,\phi}^{(k)}} + \Theta) \tag{6}$$

After the local training, MEC-DUs generate model updates and securely transmit these updates to the MEC-CU. We define χ_α as the computing time for weight aggregation and χ_β as the time for uploading and broadcasting of models during the global aggregation stage, respectively. For each FSL communication round, the training time $\Delta^{(k)}(\phi)$ can be expressed as:

$$\Delta^{(k)}(\phi) = \Delta_\mu^{(k)}(\phi) + \chi_\alpha^{(k)} + \chi_\beta^{(k)} \tag{7}$$

We need to minimize the whole training time to guarantee the FSL process runs well in the coverage of the gNodeB and improve the training efficiency. Thus the optimization problem can be expressed as:

Algorithm 1: DRFSL algorithm for multi-modal beamforming at the training stage.

Input:
$D_\phi : \{X_{\phi,C}, X_{\phi,I}, X_{\phi,\zeta}\}$ multimodal training data
Output:
$\theta_g^{FN(K)}$: Global optimized model

1 $\theta_{\phi,e}^{FN} \leftarrow \theta_g^{FN(0)}$
2 **for** $k \in \mathbb{K}$ **do**
3 **for** $j \in \mathbb{J}$ **do**
4 **for** $\phi \in \Phi$ **do**
5 $\delta_\phi^{(k)}, V_\mu \leftarrow f_{\theta_\phi}^{(k)}(L_i^\psi[1 : \mu_{\phi,\psi}^{(k)}])$
6 $\Delta^{(k,j)} \leftarrow \delta'^{(k)}_\phi \leftarrow f'^{(k)}_{\theta_\phi}, B^{(k)}$
7 **for** $e \in \mathbb{E}$ **do**
8 $\delta_e^{(k)} \leftarrow f_{\theta_e}^{(k)}(L_i^\psi[\mu_{\phi,\psi}^{(k)} + 1 : N_\psi] + L^\varpi; V_\mu)$
9 $\Theta \leftarrow w_{\phi,e}^{(j)} \leftarrow w_{\phi,e}^{(j-1)} - \frac{\eta}{\varsigma} \sum_{i=1}^{\varsigma} \nabla F_{\phi,e}(w_{\phi,e}^{(j-1)})$
10 $\delta'^{(k)}_e, V'_\mu \leftarrow f'_{\theta_e}$
11 $OptimizeMod() \leftarrow \{\Delta^{(k)}(\phi), B^{(k)}, \rho_{\phi,e}\}$
12 $Upload() \leftarrow \{\theta_\phi^{(k)}, \theta_e^{(k)}\}$
13 $\theta_g^{FN(k+1)} \leftarrow \sum_{\phi\in\Phi} \left|\frac{d_\phi}{D}\right| \theta_{\psi,\varpi}^{(k)}(\Gamma(k))$
14 **if** $k < K - 1$ **then**
15 $\mu_\phi^{(k+1)} \leftarrow OptimizeMod()$
16 $\theta_\phi^{(k)}, \theta_e^{(k)} \leftarrow Distribute(\mu_\phi^{(k+1)})$

$$\mu^* = \arg\min_{\mu} \sum_{k=1}^{K} \max_{\phi\in\Phi} \{\Delta^{(k)}(\phi)\} \tag{8}$$

where $\mu^* \in \mathbb{R}^{|\Psi|\times|\Phi|\times K}$ is the matrix of all the strategies during the FSL training. The optimization space for μ includes a cross-product of all possible combinations of OPs in $(\Gamma_C, \Gamma_I, \Gamma_\zeta, \Gamma_\varpi)$ and the corresponding set of vehicles for all rounds $\Phi \times K$. The flow of the DRFSL training framework is illustrated in Algorithm 1.

DRFSL facilitates collaborative learning between vehicles and their nearby MEC-DU servers by transmitting smash data and its gradients under dynamic network conditions. By leveraging multiple MEC-DU servers, DRFSL achieves parallel synchronization delays, outperforming sequential learning methods (refer to lines 3 to 10). Experiences are stored in the $OptimizedMod()$ for off-policy DRL training. Vehicles and MEC-DUs distributedly upload parts of their models, which the MEC-CU then aggregates based on a selective strategy. During FSL rounds, $OptimizedMod()$ aids in making decisions for mixed split policies, optimizing both computational and communication costs and distributing model parts accordingly (refer to lines 11 to 16).

2) Inference Phase: In the Inference phase, the local vehicles will receive the globally trained model from the MEC-CU. We also want to explore how our adaptive framework works well on the inference stage during the time period T. Compared with the training stage, one-time forward propagation is used for each test sample, the time of which $\Delta_\Upsilon^{(t)}(\mu_\phi)$ can be expressed as:

$$\Delta_\Upsilon^{(t)}(\mu_\phi) = (\delta_\phi + \delta_e + \frac{V_\mu(\mu_\phi^{(t)})}{B_{\phi,e}^{(t)}}) \tag{9}$$

Considering the time slot of the inference as t, we want to minimize the whole inference time for each vehicle:

$$\mu_\phi^* = \arg\min_{\mu_\phi} \sum_{t=1}^{T} \Delta_\Upsilon^{(t)}(\mu_\phi) \tag{10}$$

C. DRL-based Optimization Module

In the optimization module, we introduced DRL to assist segmentation decision-making, while the process is shown in Fig. 3. We obtain different learned DRL agents for the training and inference phases since the training phase involves heavier computing tasks and takes significantly more time.

According to the available wireless throughput and the heterogeneity of the vehicles, to use DRL algorithms, we need to define the state, action, and reward function, which are indicated as follows:

- State: s_ϕ^t represents the state of $\phi - th$ agent at section t ($t = k$ during training), which is expressed as:

$$s_\phi^t = \{\Delta_\phi^t, B_\phi^t, \mu_\phi^t, \rho_{\phi,e}^t\}_{\phi\in\Phi} \tag{11}$$

- Action: We only consider the valid partition points of the sub-model in the fusion network. a_ϕ^t represents the action of $\phi - th$ agent for the mixed OP policy contributed to the optimize module, which is expressed as:

$$a_\phi^t = \left\{ \begin{array}{ccccccc} 0 & 0 & ... & 1 & ... & 0 \end{array} \right\}_{\psi\in\Psi}, \left\|a_{\phi,\psi}^t\right\| = 1 \tag{12}$$

- Reward: $r_\phi^t(s_\phi^t, a_\phi^t)$ represents the instantaneous reward after agent ϕ at state s_ϕ^t executes action a_ϕ^t and interacts with the environment, which can be expressed as:

$$r_\phi^t(s_\phi^t, a_\phi^t) = \alpha_1(1 - \frac{T_\phi^t}{T_r}) + f(B_\phi^t; \mu_\phi^t; \rho_{\phi,e}^t) \tag{13}$$

where α_1 is the reward scale, T_r is the reference training or inference time. If the training or inference time is larger than the reference time, the first part of the reward function will get negative values, whereas it will get positive values to encourage policies for less time, and $f(\cdot)$ is the normalization of the inner product among network throughput, mixed split policy, and the capability ratio which fully deal with the heterogeneity of the edge-vehicle and the dynamic network conditions.

In the DRFSL framework, we adopt D3QN, combining advanced techniques such as double Q-learning (DDQN) [18] and dueling networks [19]. DDQN [18] addressed the overestimation decoupling the selection and evaluation of actions using the separate primary network and target network, where

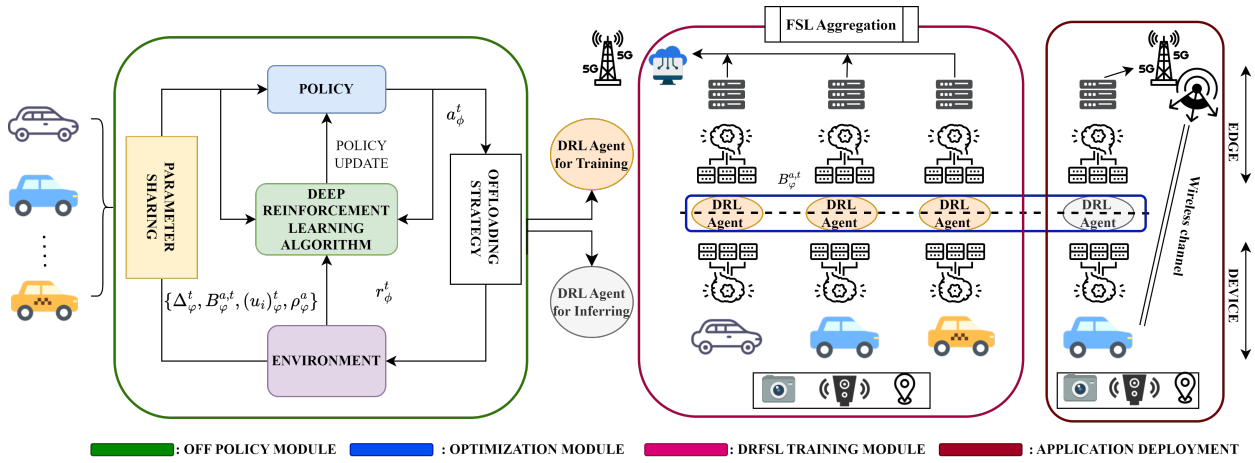

Fig. 3: Optimization module in DRFSL framework for Multi-modal beamforming in IoV

the loss function can be computed as:

$$\nabla_\theta L(\theta) = \mathrm{E}[r + \gamma Q_{\bar{\theta}}(s^{t+1}, \arg\max_{a^{t+1}}(Q_\theta(s^{t+1}, a^{t+1}))) - Q_\theta(s^t, a^t)]^2 \tag{14}$$

where γ is the discount factor, θ and $\bar{\theta}$ are the prime and the target network parameters, respectively. The dueling network [19] obtains better policy evaluation by separately estimating the state value function and the advantage function, where the Q function can be updated as:

$$Q(s^t, a^t; \theta, \alpha, \beta) = V(s^t; \theta, \beta) + (A(s^t, a^t; \theta, \alpha) - \frac{1}{\|A\|}\sum_{a'} A(s^t, a'; \theta, \alpha) \tag{15}$$

where α and β are network parameters of the different streams of the fully connected layers, $V(\cdot)$ is a state-value scalar, and $A(\cdot)$ is the advantage function. The algorithm of the DRL-based strategy at the training stage is shown in Algorithm 2.

The D3QN_Agent collects experiences as state-action-reward-next_state tuples (refer to lines 4 to 8) and updates the prime network's Q-values by minimizing loss through mini-batch sampling, utilizing a dueling architecture. Additionally, soft updates are applied at each gradient step to transfer parameters from the double Q-network to the double target network (refer to lines 9 to 13).

Algorithm 2: Training D3QN-assisted agent for multi-modal offloading policies in FSL process.

1 **Initialization:** $\overline{\theta_1} \leftarrow \theta_1, \overline{\theta_2} \leftarrow \theta_2, D \leftarrow \emptyset$
2 **for** $k \in \mathbb{K}$ **do**
3 **for** $t \in \mathbb{T}$ **do**
4 $s_\phi^t \leftarrow env()$
5 $a_\phi^t \leftarrow \varepsilon - greedy$
6 $s_\phi^{t+1}, r_\phi^t \leftarrow env(a_\phi^t)$
7 $s_\phi^{t+1} \sim p(s_\phi^{t+1}|s_\phi^t, a_\phi^t)$
8 $D \leftarrow D \cup \{(s_\phi^t, a_\phi^t, r(s_\phi^t, a_\phi^t), s_\phi^{t+1})$
9 **if** $len(D) > batch\_size$ **then**
10 $\nabla L(\theta) \leftarrow Losscompute()$
11 **for** *each gradient step* **do**
12 $\theta_m \leftarrow \theta_m - \lambda_Q \hat{\nabla}_{\theta_m} J(\theta_m), m \in \{1, 2\}$
13 $\overline{Q_m} \leftarrow \tau Q_m + (1-\tau)\overline{Q_m}, m \in \{1, 2\}$
14 **return** D3QN_Agent(θ_1, θ_2)

III. Performance Evaluation

A. Experimental Setup

All experiments, including the benchmarks, were conducted on our laboratory server which is equipped with NVIDIA A40 GPUs delivering 149.6 TFLOPS for mixed-precision (FP16/FP32) tensor operations. To simulate the computing power differences between the MEC-DU and the local vehicle, we established upper and lower capability ratio limits and uniformly sampled from these during the experiment.

1) Benchmarks: We select REFSL, which randomly samples the mixed split policy from a preset and filtered pool of behaviors, and FLASH [10] which uses FedAvg, where the training before the global aggregation is executed completely locally. Moreover, the inference time by FLASH is smaller than the beam sweeping time regulated by the IEEE 802.11ad.

2) Datasets: We assessed our experimental results using two real-world datasets: the FLASH dataset containing 25456 samples used for training and the 5G dataset used for inference [20], both relevant to moving vehicles in 5G networks. For local training, we set the mini-batch size to 32 and conducted 10 epochs for each local training session during the FSL process. The 95% confidence intervals in the result figures are calculated using the t-distribution.

B. Results Analysis

Varying network conditions: During training with an average capability ratio of 3, the DRFSL framework leverages a learned DRL agent to dynamically select optimal mixed split policies, adapting to network conditions as shown in Fig. 4. As network throughput increases, transmission of smashed data accelerates, reducing training time and boosting efficiency. At 100 MB/s, DRFSL reduces training time by up to 56.24%,

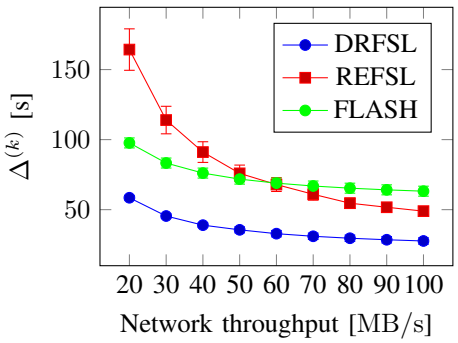

Fig. 4: Training time under dynamic network conditions.

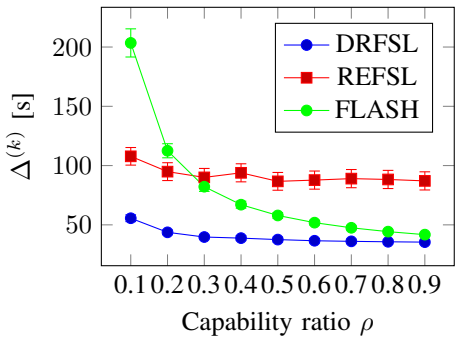

Fig. 5: Training time under different capability ratios.

from 63.19 s to 27.65 s per round compared with the standard FL (FLASH). While REFSL may outperform FLASH under ideal conditions, its unstable decisions risk disrupting global aggregation and degrading overall FSL performance.

Varying capability ratio: To account for heterogeneous vehicle computing power, we fixed the average network throughput at 40 MB/s. As shown in Fig. 5, SL becomes highly effective when the MEC server's computing power is at least twice that of the vehicle. In such cases, collaborative training significantly reduces latency. For instance, with a capability ratio of 0.2 (i.e., the server is 5× more powerful), DRFSL cuts training time to 43.70 s versus 112.50 s with FLASH. Efficiency gains range from 35.13% to 61.15% when server power is 2–5× that of vehicles, even under limited network conditions. While REFSL shows some benefit when server power exceeds 5×, it remains prone to poor policy decisions, which can severely degrade system performance.

Tradeoff between training time and accuracy In time-sensitive situations, it is crucial to complete the training task within a specified timeframe and assess the model's accuracy within that period. We evaluate our framework by accounting for the heterogeneous computing power of MEC servers and vehicles, as well as the dynamic network conditions. S1 represents the local training stage, whereas S2 denotes the aggregation stage. As shown in Fig. 6(a), our proposed DRFSL (**Blue Line**) demonstrates varying degrees of time improvement across different communication rounds compared to the FLASH (**Green Line**) using standard FL. Although REFSL (**Red Line**) occasionally performs well, it generally takes about 120% longer than the FLASH framework across all communication rounds. Within the same timeframe in Fig. 6(b), DRFSL achieves higher accuracy. For instance, around 50 minutes of training time, DRFSL attains an accuracy of 68.54%, whereas FLASH only reaches 59.84%. In Fig. 6(c), to achieve a target accuracy of over 74%, DRFSL requires 90.28 minutes, while FLASH needs 187.95 minutes. Typically, REFSL underperforms compared to FLASH, highlighting the necessity of an effective and optimal hybrid split strategy for the system. It is important to note that the training time refers solely to the duration of ML model training and excludes time for multi-sensor data collection, preprocessing, storage, and other processes. Moreover, our framework shortens response time due to reduced latency from MEC-DU to MEC-CU in the aggregation phase by the distributed uploading and broadcasting models, compared with sending the whole model directly from the vehicles.

Resilient inference stage: We tested our proposed framework using a real-world 5G dataset [20] during the inference stage. Our evaluation focused on a moving car scenario encompassing urban areas and obtained Fig. 7. Given that the inference time for a single sample is very short, REFSL for the offloading point is no longer suitable where an ineffective partitioning method can significantly harm the system. Due to the small size of the intermediate activations, the dynamic network conditions have little impact, and performance is basically related to the computing power. Our proposed DRFSL framework effectively reduces the inference time to approximately 0.2680 seconds on average, compared to 0.3547 seconds on average with the FALSH framework, which represents a 24.43% improvement on average in efficiency. It further illustrates our framework's enhanced scalability, adaptability, and robustness.

IV. CONCLUSIONS

In this paper, we propose the DRFSL framework for distributed model training in multi-modal beamforming, addressing device heterogeneity and dynamic network conditions. DRFSL utilizes MEC servers to enable efficient peer-to-peer communication and task distribution, improving both system reliability and performance. Experiments on real-world datasets demonstrate its effectiveness, with DRFSL achieving a 49.45% improvement in training efficiency and a 24.43% gain in inference performance compared to the FLASH framework while also delivering higher accuracy within the same time budget. These results highlight DRFSL's ability to manage heterogeneous resources and adapt to fluctuating network environments in multi-modal beamforming. Future work may explore integrating DRFSL with model pruning techniques to further enhance distributed learning and inference efficiency.

V. ACKNOWLEDGMENT

This research is supported by the Swiss National Science Foundation (SNSF) [Grant No. 219330] and partially supported by the U.S. National Science Foundation under grant ECCS 2516080.

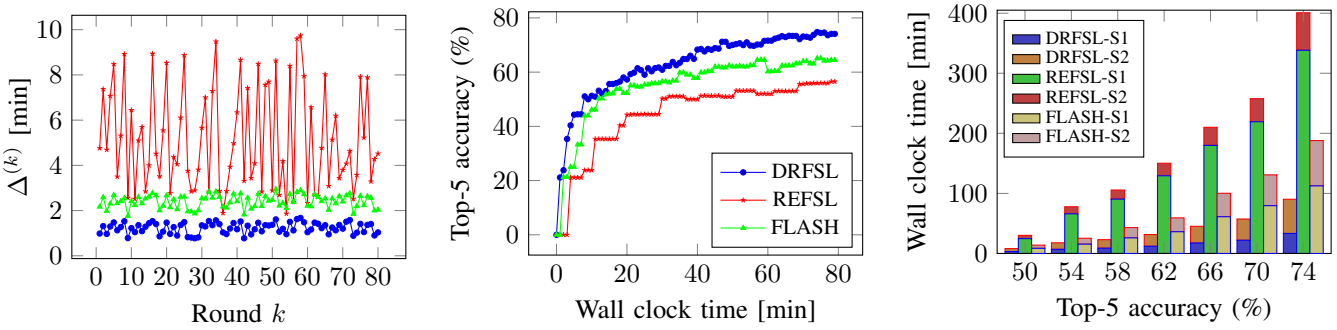

Fig. 6: (a) Training time of different methods across FSL round k, (b) Top-5 accuracy across wall clock time for different methods, (c) Wall clock time including $\Delta_{\mu}^{(k)}$ and $\chi_{\alpha}^{(k)} + \chi_{\beta}^{(k)}$ across Top-5 accuracy for different methods.

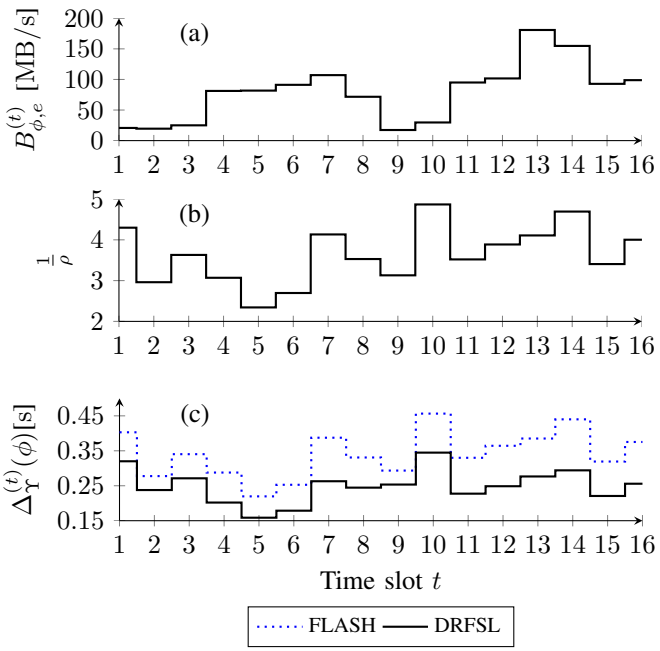

Fig. 7: The (a) real-time network throughput $B_{\phi,e}^{(k)}$, the (b) capability ratio $\frac{1}{\rho}$, and the (c) corresponding inference time $\Delta_{\Upsilon}^{(t)}(\phi)$ of DRFSL compared with FLASH at each time slot t in the dynamic IoV environment.

REFERENCES

[1] F. A. Butt, J. N. Chattha, J. Ahmad, M. U. Zia, M. Rizwan, and I. H. Naqvi, "On the integration of enabling wireless technologies and sensor fusion for next-generation connected and autonomous vehicles," *IEEE Access*, vol. 10, pp. 14 643–14 668, 2022.

[2] H. Ouamna, Z. Madini, and Y. Zouine, "6g and v2x communications: Applications, features, and challenges," in *2022 8th International Conference on Optimization and Applications (ICOA)*. IEEE, 2022, pp. 1–6.

[3] T. Shimizu, V. Va, G. Bansal, and R. W. Heath, "Millimeter wave v2x communications: Use cases and design considerations of beam management," in *2018 Asia-Pacific Microwave Conference (APMC)*. IEEE, 2018, pp. 183–185.

[4] S. S. Sarma and R. Hazra, "Pathloss attenuation analysis for d2d communication in 5g mmwave networks," in *2020 Advanced Communication Technologies and Signal Processing (ACTS)*. IEEE, 2020, pp. 1–6.

[5] J. Li, L. Xiao, X. Xu, and S. Zhou, "Robust and low complexity hybrid beamforming for uplink multiuser mmwave mimo systems," *IEEE Communications Letters*, vol. 20, no. 6, pp. 1140–1143, 2016.

[6] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave v2i beam alignment," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4042–4058, 2017.

[7] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter wave base stations with cameras: Vision-aided beam and blockage prediction," in *2020 IEEE 91st vehicular technology conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[8] A. Klautau, N. González-Prelcic, and R. W. Heath, "Lidar data for deep learning-based mmwave beam-selection," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 909–912, 2019.

[9] 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.

[10] B. Salehi, J. Gu, D. Roy, and K. Chowdhury, "Flash: Federated learning for automated selection of high-band mmwave sectors," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022, pp. 1719–1728.

[11] E. Samikwa, A. Di Maio, and T. Braun, "Disnet: Distributed micro-split deep learning in heterogeneous dynamic iot," *IEEE internet of things journal*, 2023.

[12] 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, no. 8, 2022, pp. 8485–8493.

[13] E. Samikwa, A. Di Maio, and T. Braun, "Ares: Adaptive resource-aware split learning for internet of things," *Computer Networks*, vol. 218, p. 109380, 2022.

[14] P. Ranaweera, A. K. Yadav, M. Liyanage, and A. D. Jurcut, "A novel authentication protocol for 5 g gnodebs in service migration scenarios of mec," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[15] D. Wu, R. Ullah, P. Harvey, P. Kilpatrick, I. Spence, and B. Varghese, "Fedadapt: Adaptive offloading for iot devices in federated learning," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 20 889–20 901, 2022.

[16] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.

[17] L. Lockhart, P. Harvey, P. Imai, P. Willis, and B. Varghese, "Scission: Performance-driven and context-aware cloud-edge distribution of deep neural networks," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2020, pp. 257–268.

[18] H. P. Van Hasselt, A. Guez, M. Hessel, V. Mnih, and D. Silver, "Learning values across many orders of magnitude," *Advances in neural information processing systems*, vol. 29, 2016.

[19] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[20] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: A 5g dataset with channel and context metrics," in *Proceedings of the 11th ACM multimedia systems conference*, 2020, pp. 303–308.