This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

FLASH-and-Prune: Federated Learning for Automated Selection of High-band mmWave Sectors using Model Pruning

Batool Salehi*, Debashri Roy†, Jerry Gu*, Chris Dick‡, and Kaushik Chowdhury*
*Northeastern University, USA, ‡The University of Texas Arlington, USA, and ‡Nvidia Inc.
E-mail: *{bsalehihikouei, droy, jgu1, krc}@ece.neu.edu, ‡ debashri.roy@uta.edu, ‡cdick@nvidia.com

Abstract—Fast sector-steering in the mmWave band for vehicular mobility scenarios is a challenge because standard-defined exhaustive search over predefined antenna sectors cannot be assuredly completed within short contact times. This paper proposes machine learning to speed up sector selection using data from multiple non-RF sensors, such as LiDAR, GPS, and camera images in the mmWave radios with large codebooks. The contributions in this paper are threefold: First, we propose a multimodal deep learning architecture that fuses the inputs from these data sources and locally predicts the sectors for best alignment at a vehicle. Second, we propose FLASH-and-Prune, which combines the knowledge from multiple vehicles by aggregating the local model parameters and exploits model pruning to optimize the model parameter exchange overhead. Third, we present a pruning strategy that takes into account the distributed nature of federated learning to adaptively prune or retrieve model weights. We validate the proposed architecture on a real-world multimodal dataset collected from an autonomous car. We observe that FLASH-and-Prune incurs 29.25% and 35.89% less overhead in the uplink and downlink, respectively, compared to standard federated learning.

Index Terms—sector selection, mmWave, multimodal non-RF data, federated learning, fusion, pruning.

1 Introduction

Autonomous cars are equipped with multiple sensors that stream high volumes of locally recorded data to a central cloud, which requires multi-Gbps transmission rates [1]. This data is needed for safety-critical tasks such as enhanced situational awareness, driving directives generation, and pedestrian safety, and may involve further processing at a mobile edge computing (MEC). Given the limited bandwidth in the sub-6 GHz band, the millimeter-wave (mmWave) band is an ideal candidate for vehicle-to-everything (V2X) communications [2]. As an example, emerging standards offer up to 2 GHz wide channels within the untapped spectrum resources at the 57-72 GHz.

To fully unlock the potential of mmWave-band operation, directional antennas are used to address the severe attenuation and penetration loss that is characteristic of high frequency transmissions [3]. Such antenna arrays manipulate steering directivity during runtime by changing the gain and phase of each antenna element [4]. An exhaustive search of all possible configurations results in a large overhead. Hence, current standards, such as IEEE 802.11ad and 5G-NR, prescribe a set of predefined patterns, referred to as *sectors* [5], with a deterministic sweeping algorithm that selects the optimal sector with the strongest mmWave link between transmitter (Tx) and receiver (Rx). In particular, these standards propose an exhaustive search of all sectors. This process is time-consuming as it involves probing each sector through a bi-directional packet exchange, especially for mobility scenarios where the optimal sectors may dynamically change and large codebooks.

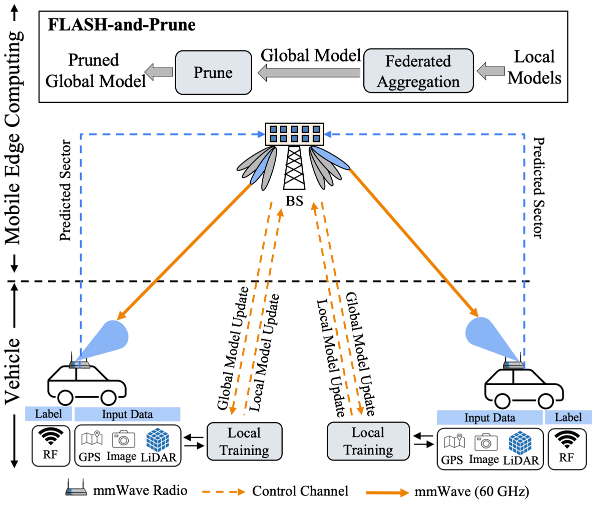


Fig. 1: The schematic of the FLASH-and-Prune framework for mmWave vehicular networks, where each vehicle is equipped with GPS, LiDAR and camera sensors. Our design employs federated learning and model pruning to aggregate the knowledge from all vehicles while minimizing the overhead caused by model parameter exchange in the control channel.

1.1 Sector Selection using Multimodal Data

Due to the quasi-optical behavior of propagation in the mmWave band, the sector selection process solves the problem of locating the strongest signal for line of sight (LOS) paths, or detecting the strongest reflection for non-line of sight (NLOS) paths. Thus, the locations of the Tx,

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

Rx, and potential obstacles play an important role in the sector selection process [6], [7]. Interestingly, all of this information is also embedded in the situational state of the environment that is acquired through monitoring sensor devices such as GPS (Global Positioning System), cameras, and LiDAR (Light Detection and Ranging), which provides a 3-D mapping of the surroundings [8], [9], [10].

Fig. 1 shows our scenario of interest with multiple moving vehicles and a roadside base station (BS) attempting to find the best sector for the downlink transmission from the BS to the vehicle. We note that multiple sensors are now included as standard installations both in modern cars and roadside infrastructures: LiDAR and camera sensors are already indispensable parts of modern vehicles, used for driving corrections and collision avoidance [11]; GPS data is regularly collected and transmitted as part of basic safety messages in V2X applications [12]. Thus, we propose a deep learning (DL) framework that uses the non-RF sensor data to select the best sector to probe *without* attempting an exhaustive search. We use a fusion architecture operating on all these different modalities to predict the optimum sector. Note that using a multitude of sensor modalities improves the prediction performance by providing a comprehensive representation of the environment. Once the best sector is determined, the BS starts the multi-Gbps downlink transmission to the vehicle, instantaneously.

1.2 Federated Learning on Multiple Modalities

DL architectures benefit from the availability of large amounts of data. When data is collected by an individual vehicle for local training, the accuracy of the model, a Deep Neural Network (DNN), may be impacted due to a limited training dataset that may not capture the diversity of other practical deployment scenarios [13]. Moreover, the vehicles must have the latest trained models available on-board when entering the network, which is difficult to accomplish without a framework for model sharing.

A federated learning (FL) architecture is one candidate solution to mitigate these issues. In this form of learning, local network models are collected from the vehicles, aggregated to a shared *global model* at the MEC, and then disseminated back to the vehicles to be used in the next iteration. Thus, vehicles collaboratively participate in learning the shared prediction model while keeping the raw training data in the vehicles instead of requiring the data to be uploaded and stored on a central server. This process is important for high-speed vehicular scenarios, as locally trained models can be updated on hidden obstacles and the unseen environment previously detected by other vehicles. Such a distributed FL architecture also allows the most updated models to be available to new vehicles that are entering the network environment. We assume that each vehicle has the necessary computation power to train and infer local machine learning (ML) models, and refer to such vehicles as *semi-autonomous edge* nodes, distinguishing them from the MEC. Moreover, we use a control channel (home WiFi or cellular network) to relay the model updates.

1.3 Federated Learning with Model Pruning

DL architectures are typically over-parameterized with much larger fitting capacities than required [14]. These unnecessary weights are not desirable for resource-constrained edge devices, where the inference is expected to be done in real-time. On the other hand, although federated learning drastically decreases the overhead by eliminating sharing of local training data, it still requires exchanging the model parameters in both uplink and downlink, periodically. As a result, having more model parameters translates to consuming more communication resources.

To further reduce the overhead of exchanging the model parameters, we propose FLASH-and-Prune, which applies model pruning after aggregation to generate a *pruned global model*. In general, pruning mechanisms remove unnecessary weights in the model to increase inference speed and decrease model storage size. However, they are typically applied to the ultimate trained models. In particular, pruning in federated learning suffers from significant challenges due to the distributed nature of this learning framework. The first challenge is to ensure the same accuracy as the original, i.e., unpruned model, while removing weights. Second, the pruning method must adapt to the knowledge learnt over federated iterations and dynamically adjust the model. The last challenge is addressing the possibility of decreasing the convergence rate that arises from diminished learning capacity caused by removing weights. Eventually, this low convergence rate increases the total number of federated iterations and might result in a much higher accumulated overhead. To account for the above challenges, we adopt a pruning algorithm tailored to the federated learning framework to optimize the parameter exchange overhead and communication efficiency, while granting the same prediction accuracy.

1.4 Our Contributions

Our main contributions are as follows:

- We design robust DL fusion architectures that predict the best sector using non-RF sensor data from devices such as GPS, camera, and LiDAR, wherein the processing steps are contained within the semi-autonomous edges (vehicles).
- We propose FLASH-and-Prune, a multimodal FL framework, where 1) the local DL model weights are globally optimized by aggregating them at the MEC instead of submitting the local sensor data, and 2) a model pruning algorithm is employed to further reduce the overhead in both uplink and downlink. Our results demonstrate that FLASH-and-Prune retains the same accuracy (∼77%) as a standard FL framework, without model pruning, while reducing the communication overhead by 29.25% and 35.89% in the uplink and downlink, respectively.
- We present a pruning strategy that is tailored to the distributed nature of the FL framework. FLASH-and-Prune takes into account loss reduction over federated iteration and relative model size to strategically select the weights to be pruned. Moreover, it can retrieve the pruned weights to maintain the accuracy as the knowledge unfolds in each iteration, if required. We compare the performance of FLASH-and-Prune against two state-of-the-art pruning strategies (iterative pruning [15] and SNIP [16]) and note up to

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

a 30.94% improvement in accuracy; thus, implying the importance of our proposed pruning strategy.

- We rigorously analyze the end-to-end latency of the FLASH-and-Prune framework and compare it with both IEEE 802.11ad and 5G-NR standards. While FLASH-and-Prune might not outperform the exhaustive search based methods in small codebooks, we make a case for using FLASH-and-Prune with large mmWave codebooks.

2 Related Works

2.1 Sector Selection via Auxiliary Information

Steinmetzer *et al.* [5] propose a compressive path tracking algorithm where the measurements on a random subset of sectors are used to estimate the optimum sector. In [17], Palacios *et al.* leverage the coarse received signal strength to extract full channel state information (CSI) and account for the overhead imposed by sector training. Saha *et al.* [18] present a comprehensive analysis of practical measurements on two commercial off-the-shelf (COTS) devices and explore the trade-off between training overhead and sector selection accuracy. Sur *et al.* [19] propose to exploit the CSI at sub-6 GHz band to infer the optimum sector at mmWave band, though it does not support simultaneous beamforming at both the Tx and Rx.

With regard to ML-based approaches, Va *et al.* [20] use the location of the target Rx as the input for their sector inference algorithm, while Alrabeiah *et al.* [21] combine both camera images and a recorded sequence of previous sectors to model mmWave communication. Klautau *et al.* [22] and Dias *et al.* [23] propose to reduce the sector search space using GPS and LiDAR sensors in vehicular settings. On the other hand, Muns *et al.* [24] use GPS and camera images to speed up the sector selection. Nevertheless, none of this literature considers real-world experiments on live sensor data. Moreover, all of the above techniques focus on a centralized system with the challenge of high bandwidth data transfer through a control channel, which is susceptible to saturation and malicious degradation.

2.2 Approaches for Efficient Federated Learning

Although FL provides frameworks to overcome the security risks with a reduced overhead [13], recent works attempt to reduce such overheads further [25]. There are efforts on decreasing the FL overhead by optimizing parameters such as the number of participating clients or the number of local iterations [26], [27]. Yang *et al.* [28] target to solve the learning and communication problem jointly by formulating an optimization problem where the goal is to minimize the total energy consumption of the system under a latency constraint. An iterative algorithm is then proposed to derive closed-form solutions for computation and transmission resources, at each federated iteration. Moreover, with respect to pruning methods, Xu *et al.* [29] employ a one shot pruning at iteration 0 followed by a selective client selection to reduce the overhead. Finally, Jiang *et al.* propose PruneFL [30] that adapts the model size during FL iterations to reduce both communication and computation overhead and minimize the overall training time. To summarize, these prior works have not investigated the effect of pruning or FL in a multimodal sensing environment. In this paper, the proposed architecture both handles challenges and validates over a live dataset. We also thoroughly study the overhead considering the growing adoption of the 5G standard.

3 System Architecture

In this section, we first review classical sector initialization and formally declare the sector selection problem. We then introduce the system architecture in FLASH-and-Prune that uses non-RF data from multiple sensors for sector selection. We summarize the notations in Table 1.

3.1 Traditional Sector Initialization

Both the IEEE 802.11ad and 5G-NR standards exploit an exhaustive search to identify the best sector for communication in mmWave bands. In the IEEE 802.11ad standard, two end-nodes, referred to as the *initiator* and *responder*, jointly explore different sectors in order to detect the best one. First, the initiator transmits a probe frame from each sector, while the responder listens to these frames in a quasi-omnidirectional antenna setting. This process is then repeated with the *initiator* and *responder* roles reversed [31]. The 5G-NR standard also employs a similar mechanism where the transmitter and receiver sequentially explore different sectors through the so called synchronization signal blocks (SSB). The maximum of 64 SSB can be grouped into a SS burst that allows up to 64 sectors to be swept in one SS burst. The 5G-NR standard defines that the SS burst duration be fixed to 5 ms, which is transmitted with a periodicity of 20 ms [32].

3.2 Sector Selection Problem Statement

Consider a Tx and Rx pair equipped with phased antenna arrays and predefined codebooks $C_{Tx} = \{t_1, \dots, t_{\mathcal{M}}\}$, $C_{Rx} = \{r_1, \dots, r_{\mathcal{N}}\}$ consisting of $\mathcal{M}$ and $\mathcal{N}$ elements, respectively. A total of $\mathcal{M} + \mathcal{N}$ probe frames or SSBs must be transmitted and the sector that returns the maximum received signal strength is then selected as the optimum sector. For example, the optimum sector at Tx is derived by:

$$t^* = \underset{1 \leq m \leq \mathcal{M}}{\arg\max}\, y_{t_m}, \tag{1}$$

with y_{t_m} being the observed received signal strength at the Rx side when the transmitter is configured at sector t_m.

3.3 FLASH-and-Prune

From Sec. 3.2, we note that the training time scales linearly with the number of sectors in the codebook and this can not be timely completed for a vehicular network with a high number of sectors (a large codebook). We propose a learning framework to exploit multiple sensor measurements available at the vehicle to locally infer the best sector t^* in one shot and then immediately start the transmission. We consider beamforming at the BS and omni-directional transmission at the vehicle. Our FLASH-and-Prune framework consists of the following steps during training and inference (see Fig. 2).

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

3.3.1 *Asynchronous Training Phase:*

The vehicles and MEC reiterate the following steps until convergence to obtain the final model.

- **Data Acquisition:** The vehicles record multimodal sensor data, including GPS, camera, and LiDAR. Moreover, the vehicles record the RF ground-truth by performing an exhaustive search (see Sec. 3.2) to obtain the optimum sector. For LiDAR preprocessing, we employ a quantization technique that incorporates the BS and vehicle position to mark the transmitter and target Rx in point clouds and the remaining detected objects as obstacles; see Sec. 4.1.
- **Local Training at the Semi-autonomous Edge:** Given preprocessed multimodal sensor data and RF ground-truth, we design a novel fusion architecture that is trained over local data (i.e., the data available at a given vehicle or each semi-autonomous edge); refer to Sec. 4.2.
- **Federated Aggregation at MEC:** The vehicles share the locally trained models with MEC in the uplink using the control channel. In FLASH-and-Prune, we use a buffered asynchronous model aggregation scheme [33], [34]. Thus, the model updates are sent asynchronously over time, when the vehicles are stationary (for example at the end of the trips where they could simply use their home WiFi). These model updates are stored in a buffer and the MEC performs aggregation once it receives updates from a few vehicles (once a day for example). Given the local model updates from participating vehicles, we propose a multimodal FL-based architecture to aggregate the local model updates and attain a *global model*; see Sec. 4.3. The buffered asynchronous aggregation strategy ensures that sharing the model weights in FLASH-and-Prune does not affect the performance of the other regular users.
- **Pruning for Multimodal Federated Learning:** We strategically prune the least significant weights from the global model to generate the *pruned global model*. The MEC then reports back the pruned global model in the downlink using the control channel, which is used by vehicles in the next federated iteration; refer to Sec. 4.4. Similar to the previous step, the global model update is shared at the beginning of each trip from the MEC to the vehicles using the home WiFi.

3.3.2 *Inference Phase:*

At the end of the training phase, the vehicles are updated with the final model for sector prediction. Thus, in the inference phase, the vehicles use the sensor data and run a single forward pass to predict the optimum sector. As a result, the inference happens locally at the vehicles. Our system includes sector selection at the BS. Thus, after inference, the predicted sector is shared with the BS using the control channel as the selected sector at the BS. Available interfaces such as Open Radio Access Network (O-RAN) [35] systems enable the BS to immediately plug in the inferred sector and start transmission in the mmWave band.

Notation	Description
C_{Tx}	Codebook of transmitter with $\mathcal{M}$ sectors
C_{Rx}	Codebook of receiver with $\mathcal{N}$ sectors
y_{t_m}	Received signal strength for $t_m \in C_{Tx}$
t^*	Optimum sector at Tx
$X_{\{\mathtt{C},\mathtt{I},\mathtt{L}\},\nu}$	Local samples of GPS, image and LiDAR at vehicle ν
$N_t, N_t^{'}$	Number of train and test samples
$f_{W^{\nu}_{\{\mathtt{C},\mathtt{I},\mathtt{L},\mathtt{FN}\}}}$	Unimodal and integration networks for vehicle ν
V_i	Number of participating vehicles at iteration i
$\mathcal{N}$	Number of federated iterations
W_i^{ν}	Local model weights for vehicle ν at iteration i
W_i	Global model weights at iteration i
M	Model pruning mask
$\mathcal{L}(W)$	Federate loss for weights W
$W_i^{'}$	Pruned global model weights at iteration i
$g^{\nu}_{W^{'}_{i-1}}$	Gradient at vehicle ν after local training when initialized by $W^{'}_{i-1}$
P	Parameter space of the model
M_p, M_t	Number of pruned and unpruned model parameters
$C(R)$	Relative model size

TABLE 1: Notation Summary

4 FLASH-AND-PRUNE FRAMEWORK

In this section, we describe the key components of proposed FLASH-and-Prune framework.

4.1 Data Acquisition

To process the LiDAR data, we first construct a quantized view of the spatial extent of the surroundings. This data structure resembles a stack of cuboid regions placed adjacent to each other. The LiDAR point clouds reside in the cuboid regions according to their relative distances as measured from a shared origin as in [23]. We mark the cuboids that contain blocking obstacles using label 1. Since we know the coordinates of the Tx and Rx, we label the cuboids containing them as -1 and -2, respectively. The LiDAR preprocessing happens at the vehicle. Thus, the BS must share its location in downlink with the vehicles. This can be done when the vehicles check in with the BS.

4.2 Local Training at Semi-autonomous Edge

Consider a number of vehicles V that are in the coverage range of the BS and are trying to establish a link with the latter. Each vehicle is equipped with GPS, camera, and LiDAR sensors and collects the local dataset $D_\nu = \{X_{\mathtt{C},\nu}, X_{\mathtt{I},\nu}, X_{\mathtt{L},\nu}\}_{\nu=1}^{V}$. We denote the data matrices for GPS, image, and LiDAR at the vehicle ν as $X_{\mathtt{C},\nu} \in \mathbb{R}^{N_t \times 2}$, $X_{\mathtt{I},\nu} \in \mathbb{R}^{N_t \times d_0^{\mathtt{I}} \times d_1^{\mathtt{I}}}$, $X_{\mathtt{L},\nu} \in \mathbb{R}^{N_t \times d_0^{\mathtt{L}} \times d_1^{\mathtt{L}} \times d_2^{\mathtt{L}}}$, respectively, where N_t is the number of training samples. Furthermore, $(d_0^{\mathtt{I}} \times d_1^{\mathtt{I}})$ and $(d_0^{\mathtt{L}} \times d_1^{\mathtt{L}} \times d_2^{\mathtt{L}})$ give the dimensionality of image and preprocessed LiDAR data, while the GPS has 2 elements. The label matrix $Y_\nu \in \{0,1\}^{N_t \times \mathcal{M}}$ represents the one-hot encoding of $\mathcal{M}$ sectors, where the optimum sector is set to 1, and rest are set to 0 as per Eq. (1). Each vehicle uses its local dataset D_ν to initiate a supervised learning task. In the simplest case, the vehicles uses a DNN-based unimodal network to extract discriminative features from the input and infer the optimum sector. Each unimodal

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

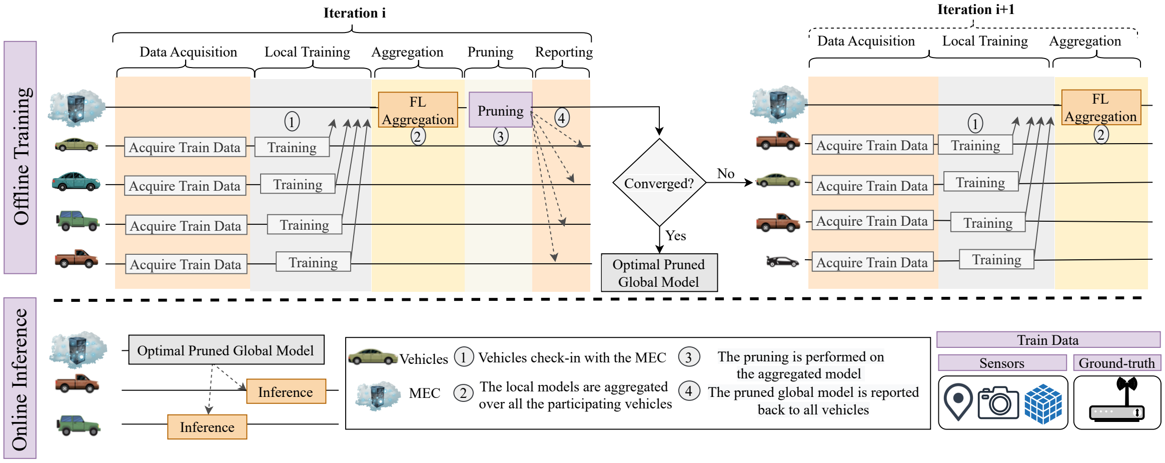


Fig. 2: Operating steps of the proposed FLASH-and-Prune architecture consisting of: (a) data acquisition, (b) local training, (c) aggregation, (d) pruning, and (e) reporting. The time window for each step depends on the application requirements.

network makes a probabilistic prediction of the best sector through softmax layer σ as:

$$\mathbf{u}^{\nu}_{\mathtt{C}} = \sigma(f_{W^{\nu}_{\mathtt{C}}}(X_{\mathtt{C},\nu})), \qquad f_{W^{\nu}_{\mathtt{C}}} : \mathbb{R}^2 \mapsto \mathbb{R}^M \tag{2a}$$

$$\mathbf{u}^{\nu}_{\mathtt{I}} = \sigma(f_{W^{\nu}_{\mathtt{I}}}(X_{\mathtt{I},\nu})), \qquad f_{W^{\nu}_{\mathtt{I}}} : \mathbb{R}^{d_0^{\mathtt{I}} \times d_1^{\mathtt{I}}} \mapsto \mathbb{R}^M \tag{2b}$$

$$\mathbf{u}^{\nu}_{\mathtt{L}} = \sigma(f_{W^{\nu}_{\mathtt{L}}}(X_{\mathtt{L},\nu})), \qquad f_{W^{\nu}_{\mathtt{L}}} : \mathbb{R}^{d_0^{\mathtt{L}} \times d_1^{\mathtt{L}} \times d_2^{\mathtt{L}}} \mapsto \mathbb{R}^M \tag{2c}$$

where $f_{W^{\nu}_{\mathtt{C}}}(.)$, $f_{W^{\nu}_{\mathtt{I}}}(.)$, $f_{W^{\nu}_{\mathtt{L}}}(.)$ denotes the unimodal network for each vehicle ν parameterized by weights $W^{\nu}_{\mathtt{C}}$, $W^{\nu}_{\mathtt{I}}$, $W^{\nu}_{\mathtt{L}}$. On the other hand, using the data from all sensing modalities can boost the prediction performance. Hence, we design a *fusion* network that consists of four DNNs, three unimodal networks (Eq. (2)), and an integration network $f_{W^{\nu}_{\mathtt{IN}}}(.)$ parameterized by $W^{\nu}_{\mathtt{IN}}$. Formally,

$$f_{W^{\nu}_{\mathtt{FN}}}(.) = f_{W^{\nu}_{\mathtt{IN}}}(f_{W^{\nu}_{\mathtt{C}}}(.), f_{W^{\nu}_{\mathtt{I}}}(.), f_{W^{\nu}_{\mathtt{L}}}(.)), \tag{3a}$$

$$\mathbf{u}^{\nu}_{\mathtt{FN}} = f_{W^{\nu}_{\mathtt{FN}}}(X_{\mathtt{C},\nu}, X_{\mathtt{I},\nu}, X_{\mathtt{L},\nu}), \tag{3b}$$

where $f_{W^{\nu}_{\mathtt{FN}}}(.)$ is the fusion model parameterized by $W^{\nu}_{\mathtt{FN}}$. The unimodal networks and integration network can be fused together following different architectural designs, such as concatenation at the penultimate layer. Finally, the prediction happens at the output of fusion network through the computation of $\mathbf{s} = \sigma(\mathbf{u}^{\nu}_{\mathtt{FN}})$. The sector that has the highest score is chosen as the predicted sector.

4.3 Multimodal Federated Aggregation

In the local training step, each vehicle ν performs training on the local multimodal data for a few epochs using the fusion network described in Sec. 4.2 and attains the local weights as $W^{\nu}_{\mathtt{FN}}$. For simplicity, in the rest of the paper, we omit the subscript FN and denote the local weights at vehicle ν at federated iteration i as W^{ν}_i. If a vehicle decides to participate, it sends the local model weights for the overall fusion network (encapsulating four branches, GPS, image, LiDAR, and integration) to the MEC, using the control channel. Moreover, since the training phase is asynchronous, the BS waits for receiving a few model updates and stores them in buffers before proceeding to the next step. In other words, the training happens over the course of time not the sector coherence time. At the aggregation step of i^{th} iteration, the MEC uses received local weights $\{W^{\nu}_i\}_{\nu=1}^{V_i}$ from V_i vehicles and employs an averaging scheme [13] to aggregate the local model weights and calculates the global model weights W_i. Formally,

$$W_i = \frac{1}{V_i} \sum_{\nu=1}^{V_i} W^{\nu}_i, \tag{4}$$

Control Channel Overhead in standard FL. In a standard FL architecture, the entire global model weights W_i are transmitted back to all vehicles and used as the initialization weights in the next federated iteration. In such case, the global optimization of the local models requires the vehicles to periodically exchange the local model parameters $\{W^{\nu}_i\}_{\nu=1}^{V_i}$ in the uplink and the MEC to report back the global model W_i to all vehicles in the downlink, in each federated iteration. These parameter exchanges impose overhead of $\widetilde{o^S_{ul}}$ and $\widetilde{o^S_{dl}}$ variables in the uplink and downlink control channels as:

$$\widetilde{o^S_{ul}} = \sum_{i=1}^{\mathcal{N}} V_i \times (|W^{\nu}_i|), \qquad \widetilde{o^S_{dl}} = \mathcal{N} \times (|W^{\nu}_i|), \tag{5}$$

where $\mathcal{N}$ is the total number of federated iterations. Note that in a standard federated learning architecture, the local and global models have the same number of parameters that is constant over iterations, $|W_i| = |W^{\nu}_i| = |W^{\nu}_{\mathtt{C}}| + |W^{\nu}_{\mathtt{I}}| + |W^{\nu}_{\mathtt{L}}| + |W^{\nu}_{\mathtt{IN}}| \quad \forall i \in \{1, \cdots, \mathcal{N}\}$. Given the depth of the DNNs, sharing all the locally trained weights for the three different unimodal and one integration models to the MEC occupies approximately 26.54 MB, which also scales with the number of participating vehicles and iterations. Thus, we propose a model pruning algorithm to reduce the number of exchanged parameters in both uplink and downlink.

4.4 Pruning for Multimodal Federated Learning:

Model pruning techniques allow real-time inference for resource-constrained devices [36], [37]. However, in distributed learning architectures such as federated learning, they may reduce the number of model parameters to be exchanged and improve the communication efficiency. In this section, we first describe the dissemination of our proposed pruning module over the federated learning framework in

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

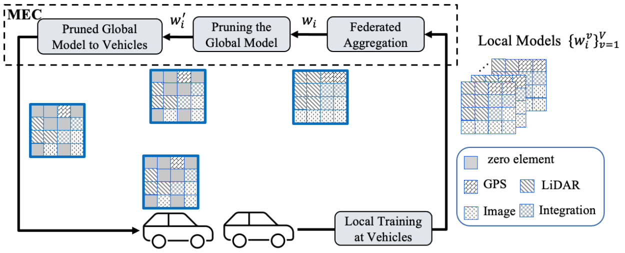


Fig. 3: Layout of FLASH-and-Prune Sec. 4.4 at training phase, where the model pruning module identifies the negligible weights throughout the fusion architecture.

FLASH-and-Prune, and then describe our pruning strategy. At the end, we discuss important design parameters and describe the FLASH-and-Prune framework end-to-end.

4.4.1 Proposed Pruning Approach

In general, a pruning algorithm identifies a sub-network of the original model, where a portion of the model weights are selected to be pruned with the goal of compressing the model, while maintaining the accuracy. The pruned model is obtained by applying a *mask* on the original model weights, with elements $M_i = \{0, 1\}$ where the 0 and 1 indices refer to the pruned and remained weights, respectively.

Types of Pruning. The model pruning strategies are either unstructured, in which individual weights are pruned, or structured, in which an entire neuron and its weights are pruned together. The structured pruning severely limits the maximum achievable sparsity in comparison with unstructured pruning. Moreover, it does not conform to the *lottery ticket* hypothesis [38] that implies: if randomly-initialized, the *winning ticket* sub-network reaches a test accuracy comparable to the original network in a similar number of training iterations. In this paper, we opt for unstructured pruning since it allows for maximum model compression.

Approach. In FLASH-and-Prune, we improve the multimodal FL framework by including the pruning over the aggregated global model, which is a multimodal architecture consisting of all four branches (GPS, LiDAR, image, and integration); see Fig. 3. Overall, the pruning module is disseminated over the multimodal federated learning framework in these following stages:

1) *Local Models to MEC:* All participating vehicles share their local models $\{W_i^\nu\}_{\nu=1}^{V_i}$ with the MEC.
2) *Federated Aggregation:* Given the local model weights from all vehicles at iteration i, the MEC aggregates weights using Eq. (4) to generate the global weights at i^{th} iteration, W_i.
3) *Pruning the Global Model:* The global weights are then passed through the pruning algorithm and generate the pruned global model weights $W_i^{'}$ by applying the pruning mask M. Formally, the pruned model weights are derived by calculating the element-wise product with the pruning mask as: $W_i^{'} = M \odot W_i$.
4) *Pruned Global Model to Vehicles:* The pruned model weights are then transmitted back to all vehicles and used as initialization for the next federated iteration.

4.4.2 Model Pruning Strategy

In FLASH-and-Prune, we choose a pruning strategy that enables us to adaptively shrink or expand the model. In the case of centralized learning, the model is pruned after training. However, in federated learning architecture, the model is still being updated over the federated iterations. Thus, the importance of weights can change drastically from one iteration to the other. Thus, having an adaptive method that can prune and also return back the weights ensures that the pruning method utilizes the learnt knowledge over federated iterations. To properly construct a pruned model in FLASH-and-Prune, we define two metrics: (a) loss reduction over federated iteration and (b) relative model size.

(a) Loss Reduction over Federated Iterations: The loss reduction metric captures the change in the global loss while going from one iteration to the next, following the adaptive pruning scheme discussed by Jiang *et al.* [30].

Theorem 1. *If $\mathcal{L}(W_{i-1}^{'}) \approx \mathcal{L}(W_{i-1})$, the loss reduction over federated iterations, $\mathcal{L}(W_{i-1}) - \mathcal{L}(W_i)$, relates to the sum of squared aggregated gradients over all vehicles after local updates, $\sum_{r\in R} g^2_{W_{i-1}^{'}}(r)$, when initialized with the pruned model from the previous iteration.*

Proof. Recall that FLASH-and-Prune runs over four steps as described in Sec. 4.4.1, where each iteration starts with running the local training at vehicles and ends with reporting back the pruned global model to vehicles. As a result, at the beginning of each federated iteration, the vehicles receive the latest pruned global model from the MEC, i.e., $W_{i-1}^{'}$. The vehicles then initialize their local models with the pruned global model ($W_{i-1}^{'}$) from previous iteration and perform local training using their local datasets D_ν. In this case, the global model in the current iteration i, after local training, reporting, and aggregation steps, is described as:

$$W_i = W_{i-1}^{'} - \eta g_{W_{i-1}^{'}} \odot M_{W_{i-1}^{'}}, \tag{6}$$

where η and $M_{W_{i-1}^{'}}$ denote the learning rate and mask of the pruned global model and $\odot$ is the element-wise product. In Eq. (6), $g_{W_{i-1}^{'}}$ is the aggregated local gradients over all vehicles, collected after local training step:

$$g_{W_{i-1}^{'}} = \frac{1}{V}\sum_{\nu=1}^{V} g^\nu_{W_{i-1}^{'}}. \tag{7}$$

In a federated learning architecture, the federated loss is denoted as the average of losses (e.g., cross-entropy, mean square error, etc.) over all clients. Formally,

$$\mathcal{L}(W) = \frac{1}{V}\sum_{\nu=1}^{V} \mathcal{L}^v(W), \tag{8}$$

where $\mathcal{L}^v(W)$ denotes the local loss at vehicle ν for weights W. We estimate the global loss in the current iteration $\mathcal{L}(W_i)$ using Taylor expansion as:

$$\mathcal{L}(W_i) = \mathcal{L}(W_{i-1}^{'}) + \langle \nabla\mathcal{L}(W_{i-1}^{'}), W_i - W_{i-1}^{'} \rangle \tag{9a}$$

$$= \mathcal{L}(W_{i-1}^{'}) - \eta \langle \nabla\mathcal{L}(W_{i-1}^{'}), g_{W_{i-1}^{'}} \odot M_{W_{i-1}^{'}} \rangle \tag{9b}$$

$$\approx \mathcal{L}(W_{i-1}^{'}) - \eta || g_{W_{i-1}^{'}} \odot M_{W_{i-1}^{'}} ||^2 \tag{9c}$$

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

where $||.||$ denotes the norm operation on matrices. In the above equation, Eq. (9b) is derived by replacing the second term in the inner product using Eq. (6) and Eq. (9c) is derived by approximating the stochastic gradient with its expectation $\nabla\mathcal{L}(W^{'}_{i-1}) \approx g_{W^{'}_{i-1}}$. As a result, the reduction in loss from the latest pruned model to the aggregated model weights in the current iteration is approximated as:

$$\mathcal{L}(W^{'}_{i-1}) - \mathcal{L}(W_i) = \eta||g_{W'_{i-1}} \odot M_{W'_{i-1}}||^2, \tag{10a}$$

$$\propto ||g_{W'_{i-1}} \odot M_{W'_{i-1}}||^2, \tag{10b}$$

$$= \sum_{r\in R} g^2_{W'_{i-1}}(r), \tag{10c}$$

where R denotes the index set of remaining components in the model, i.e., the components that are not pruned ($|R| = M_p$). Interestingly, we observe the difference between the federated loss of the pruned model from previous iteration $W^{'}_{i-1}$ and current aggregated model W_i relates to the sum of squared aggregated gradients over all vehicles after local updates, when initialized with the pruned model from the previous iteration.

However, our goal is to estimate the loss reduction from one federated iteration to the next, i.e., $\mathcal{L}(W_{i-1}) - \mathcal{L}(W_i)$. We estimate the federated loss at the iteration $i-1$, by the loss of pruned model as there is no straight way to compute it. However, to ensure that the approximation is valid, we impose a constraint on our proposed optimization problem.

$$\mathcal{L}(W^{'}_{i-1}) \approx \mathcal{L}(W_{i-1}). \tag{11}$$

Hence, from Eq. 10c and Eq. 11, we prove the Theorem. 1, by concluding:

$$\mathcal{L}(W^{'}_{i-1}) - \mathcal{L}(W_i) \propto \sum_{r\in R} g^2_{W'_{i-1}}(r). \tag{12}$$

□

(b) Relative Model Size: This parameter denotes the relative size of pruned model with respect to the original unpruned model. With M_p and M_t being the number of pruned model and original unpruned model parameters, the relative model size is defined as below:

$$C(R) = \sum_{r\in R} \frac{1}{M_t} = \frac{M_p}{M_t}, \quad M_p = |R|. \tag{13}$$

Intuitively, the relative model size provides a realization about the required wireless resources to share the model parameters.

Optimization. Note that both the above metrics are a function of the remaining neural network components R. Intuitively, the loss reduction over federated iteration and relative model size establishes a trade-off between retaining the accuracy and the required resources for exchanging the model parameters, respectively. Thus, we define the parameter Δ as the ratio of two metrics:

$$\Delta(R) = \frac{\mathcal{L}(W_{i-1}) - \mathcal{L}(W_i)}{C(R)} \propto \frac{\sum_{r\in R} g^2_{W'_{i-1}}(r)}{C(R)}. \tag{14}$$

As a result, identifying the optimum sub-network translates to maximizing the Δ over the entire network parameter space P. However, to account for the assumption in Eq. (11), we impose a constraint on the optimization problem and prevent removing the weights that are essential for maintaining the accuracy of pruned model as:

$$\max \quad \Delta(S \cup E), \tag{15a}$$

$$\text{s.t.} \quad S \subseteq \overline{E}. \tag{15b}$$

Algorithm 1: FLASH-and-Prune

```
Input: Pruned model from previous iteration $W^{'}_{i-1}$
Output: Pruned global model in current iteration $W^{'}_{i}$
At Vehicles:
Initialize local models with $W^{'}_{i-1}$
Local training for $\xi$ epochs
Collect updated local model weights $\{W^{\nu}_i\}^{V_i}_{\nu=1}$ and
  local gradients $\{g^{v}_{W'_{i-1}}\}^{V}_{\nu=1}$ for all vehicles
At MEC:
Calculate the aggregated global model $W_i$ (Eq. (4))
Calculate the aggregated gradient $g_{W'_{i-1}}$ (Eq. (7))
Construct $E$ using Eq. (16)
$S \leftarrow \emptyset$
$\delta \leftarrow \arg\operatorname{sort}_{j\in\overline{E}} \frac{g^2_{W'_{i-1}}(j)}{C(j)}$
for $j \in \delta$ do
    if $\frac{g^2_{W'_{i-1}}(j)}{C(j)} \geq \Delta(S \cup E)$ then
        $S \leftarrow S \cup j$
    else
        break
    end
end
$W^{'}_{i} = S \cup E$
MEC distributes $W^{'}_{i}$ such $W^{\nu}_i = W^{'}_i \; \forall\nu \in V$
```

In particular, we partition the parameter space P into two disjoint subsets E and $\overline{E}$, where $E \cup \overline{E} = P$. The set E denotes the essential weights that cannot be pruned to satisfy the assumption in Eq. (11). This includes weights whose magnitudes are larger than a certain threshold:

$$E = thresh(W_i) = \begin{cases} 1 & \text{if } |W_i| > \lambda \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

The set $\overline{E}$ refers to the weights that can be potentially removed. From this set, we select a subset S which corresponds to the weights that are chosen to be kept in the network. To construct S, we first calculate Δ for all the weights in the set $\overline{E}$ and sort the values in non-increasing order. We then gradually collect the weights from largest and add them to the set S. We stop when adding an individual weight does not increase the overall Δ. The selected weights in the current federated iteration are the union of the set S and E elements. Our sub-network selection algorithm is summarized in Alg. 1. Note that the set S can dynamically grow or shrink for each operation. This feature grants the flexibility to adapt the model size according to the knowledge learnt throughout the federated iterations. As denoted in [30], an adaptive pruning algorithm converges as long as the number of nonzero prunable parameters in $\overline{E}$ decreases over the iterations.

4.4.3 *Initial Pruning and Adjustment Iterations*

The optimization problem in Eq. (15) provides the optimum sub-network that learns the fastest. In order to obtain the so-

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

lution, the gradients on the full parameter space are required to compute the loss reduction over federated iterations, see Eq. (9). As a result, the clients must transmit the *gradients* along with *model weights*. It is also crucial to determine *when* and *how* these gradients along with model weights should be exchanged back and forth between the MEC and vehicles. We present a three-phase modular design that helps to regulate this exchange in the FLASH-and-Prune framework. These three phases are: (a) initial pruning at a selected client, (b) non-adjustment iterations, and (c) adjustment iterations.

(a) Initial Pruning at Selected Client: The model pruning gradually decreases the number of parameters by removing the unnecessary weights. However, to further optimize the number of parameters to be transmitted, we propose to employ an extra pruning step at a selected client before starting the federated iterations. The initial pruning steps follow the same structure as pruning strategy described in Sec. 4.4.2. This enables the federated learning to start with a smaller model at the beginning and decrease the channel occupation. Recall that the adaptive feature of the proposed pruning strategy in Sec. 4.4.2 enables handling the initial pruning step and retrieving weights, if required.

(b) Adjustment Iterations: We note that the pruning step does not need to be performed in each federated iteration as the global model requires some iterations to be significantly updated, and it also unnecessarily increases the overhead. We design the system such that the pruning step happens once in multiple FL iterations, a parameter that we refer to as *adjustment frequency* (AF). In an adjustment iteration, the MEC receives the local models and gradients from all clients. After federated aggregation, the MEC solves Eq. (15) to identify the optimum pruned model.

(c) Non-adjustment Iterations: The process of steps in a non-adjustment iteration follows the same structure as a standard FL architecture and includes three stages of local training, aggregation, and reporting (see Sec. 4.3). Note that in a non-adjustment iteration vehicles only share the model parameters to the MEC.

4.4.4 End-to-End FLASH-and-Prune Framework

The overall FLASH-and-Prune architecture is depicted in Fig. 4 and runs as follows. First, the MEC selects a client to perform the initial pruning step using its local data. The selected vehicle then shares the pruned model with the MEC that is used for initializing the global model. By starting the federated iterations, the vehicles use the local data to train the models and collect the gradients over the entire parameter space. In an adjustment iteration, the vehicles send the local model weights and gradients to the MEC. The MEC then aggregates the local models using Eq. (4) and identifies the optimum sub-network by solving Eq. (15). The pruned global model is then sent back to all vehicles. If the system is not in an adjustment iteration, the system follows the same structure as a standard FL architecture.

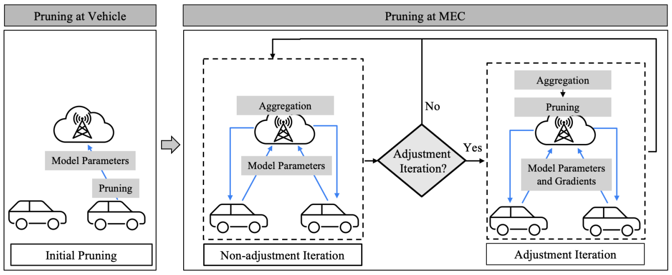


Fig. 4: Three phases of FLASH-and-Prune framework. The initial pruning happens at a selected client only once before starting the federated iterations. The pruning at MEC frequency is set according to adjustment frequency.

Control Channel Overhead. It is to be noted that the FLASH-and-Prune impacts both the uplink and downlink channel overhead following its pruning layout. Formally, the overhead in uplink and downlink is characterized as:

$$\widetilde{o^P_{ul}} = \sum_{i=1}^{\mathcal{N}} V_i \times [(1 - \mathbb{1}_{i \bmod \mathrm{AF} \neq 0}) \times |M_{i-1}| + \mathbb{1}_{i \bmod \mathrm{AF} = 0} \times (M_t + |M_{i-1}|)] \tag{17a}$$

$$\widetilde{o^P_{dl}} = \sum_{i=1}^{\mathcal{N}} |M_i|, \tag{17b}$$

where ϕ is a Boolean predicate, with $\mathbb{1}_\phi$ to be 1 if ϕ is true, and 0 otherwise, and denotes being in an adjustment iteration. Note that the first term in Eq. (17a) accounts for the overhead in the case of non-adjustment iterations where only the local weights (with the number of parameters equal to the pruned model from the previous iteration) are sent in the uplink. The latter in Eq. (17a) corresponds to an adjustment iteration where the local model weights and gradient over the entire parameter space (for all M_t parameters of the original unpruned model) are sent in the uplink. At the online inference phase, the vehicles use the optimal global model and run the inference locally online.

5 Experiments

In this section, we present an experimental multimodal dataset (named '*FLASH Dataset*') that is published in [39] for community use. We then describe our competing methods, implementation details, and performance metrics.

5.1 Dataset

The FLASH dataset is collected in the city of Boston, on a two-way paved alleyway between two high-rise buildings. A 2017 Lincoln MKZ Hybrid autonomous car is equipped with on-board GPS, GoPro HERO4 camera, and Velodyne VLP-16 LiDAR. Furthermore, two TP-Link Talon AD7200 tri-band routers (operating at 60 GHz) with 34 pre-defined codebooks are located at the road-side BS and top of the vehicle. The RF ground-truth includes the RSSI (received signal strength indicator) observed at the Rx which has omni-directional steering, while the BS is configured at each of the 34 sectors. The dataset includes the synchronized multimodal data as well as the RF ground-truth, the RSSI for each of the 34 sectors or classes. The FLASH dataset spans a variety of LOS and NLOS scenarios with pedestrians and static or moving cars as obstacles.

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

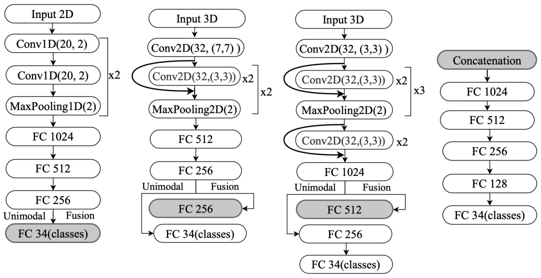
5.2 Competing Methods

We benchmark the performance of FLASH-and-Prune framework against competing methods below.

- **Local Learning:** The vehicles use their own local training data to optimize the local models, independently. In this method, there is no data sharing; vehicles operate as disjoint independent clients and the training data is confined to their own local data only.
- **Centralized Learning:** The vehicles participate in a data sharing scheme to converge to a generalized model. As a result, all vehicles transmit their own local training data that is centrally collected at the MEC. The latter trains a model on the accumulated training data. This scheme requires a control channel with the required bandwidth for sharing such large amounts of data.
- **Standard FL Architecture:** The vehicles use only their local training data to optimize their local model. Each vehicle participates in a global model aggregation iteration, where only the local models are sent to the MEC.
- **FLASH:** An orchestrator designates a branch to be aggregated at the MEC, and only the weights of the updated branch are transmitted back to the vehicles [40]. Thus, the knowledge learned by other modality is entirely discarded.
- **FLASH-and-Prune:** An extra pruning module is employed along with standard FL to further optimize the model exchange overhead. This includes initial pruning at a selected client before starting the federated iteration and further pruning at adjustment iterations at the MEC.

5.3 Implementation Details

We use the entire FLASH dataset with 4 different categories and 21 scenarios (inclusive of LOS and NLOS). Each scenario consists of 10 episodes or trials of data collection and can be interpreted as having different vehicles. In this way, we have 10 different vehicles, each having a total of 21 different scenarios as their local dataset. During the collection of the FLASH dataset, different episodes of the same scenario are designed to be different, making each local dataset (per vehicle) unique. To replicate real-world situations, we create *local training* and *validation* datasets for each vehicle by randomly separating 80% and 10% of the data in each episode. However, to expose the trained models to the unseen environment detected by other vehicles, we create a *global test dataset*, where we combine the leftover 10% of each vehicle's local data. The overall dataset contains 25456 and 3180 local training and validation and 3287 global test samples, respectively. We set the LiDAR range to be within ± 80 m. We quantize each axis to a (20, 20, 20) block array which corresponds to steps of (2.79, 4.65, 0.5). Moreover, we resize the high quality raw images to (160, 90, 3). For all models (see Fig. 5), we exploit categorical cross-entropy loss for training with a batch size of 32. For local and centralized learning, we use 150 training epochs; for FL-based methods, we use 20 local training epochs. Moreover, we use early stopping based on the validation accuracy to avoid overfitting and report the test accuracy of the best model for all competing methods. We use Adam [41] as our optimizer with $\beta = (0.9, 0.999)$ and initialize the learning rate to 0.0001. For pruning experiments, we set the λ in Eq. (16) such that the thresholding function identified top 30% weights with the highest absolute value.



(a) (b) (c) (d)

Fig. 5: Proposed network architectures for (a) GPS, (b) image, (c) LiDAR, and (d) integration networks. The integration model is designed by concatenating the highlighted layers from each unimodal model. We set the dimensionality of high level features according to the importance of each sensor modality. While the GPS data does not include significant features that could be extracted, we design our fusion network to map it to high level features with dimensionality of 32. By increasing the dimensionality of GPS data representation, we ensure that the significance of this sensor is not glossed over compared to the other sensor modalities.

5.4 Performance Metrics

The errors in prediction, i.e., selecting a sub-optimal sector, can affect the performance. Thus, we evaluate the sector prediction performance by defining throughput ratio as $R_T = \frac{1}{N_t^{'}} \sum_{n=1}^{N_t^{'}} \frac{\log_2[1+y_{\hat{t}}(n)]}{\log_2[1+y_{t^*}(n)]}$. Here, t^* and $\hat{t}$ denote the best ground-truth sector and the predicted sector, respectively, and $N_t^{'}$ is the total number of test samples. Intuitively, this metric captures the ratio of degradation in performance compared to the ideal exhaustive search method. We evaluate the model pruning performance via compression rate. Formally, for n and $n^{'}$, the total number of model parameters for the original unpruned and pruned models, respectively, the compression ratio is $\frac{n-n^{'}}{n}$.

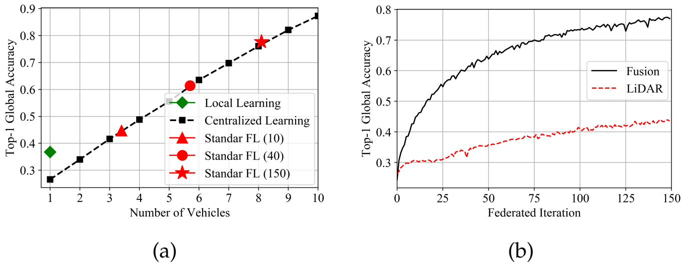
6 Evaluation of the Competing Methods

We compare the competing learning-based methods in this section. For all experiments, we use a global test set to evaluate performance.

6.1 Necessity of Federated Learning

In the first set of our experiments, we compare the accuracies with local, centralized and standard federated learning strategies in Fig. 6a. In local learning, we train DNNs on the local dataset for each vehicle and observe the top-1 accuracy range of 12%-36% over all 10 vehicles (maximum accuracy is denoted with a diamond marker in Fig. 6a). In centralized learning, we construct an *accumulated training set* by gathering the local training set at MEC. We begin with the data from a single vehicle and increase the accumulated training set by adding the local data from other vehicles, one at a time (see horizontal axis in Fig. 6a). We observe a surge in top-1 accuracies, up to up to 87.31% accuracy, as we add more vehicles to the accumulated training set at the

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.



(a)

(b)

Fig. 6: Comparing (a) the performance of standard FL with local learning and an increasing number of vehicles in centralized learning. The numbers in parenthesis denote the federated iteration (b) top-1 global accuracy for LiDAR only and fusion of all three modalities with standard FL architecture.

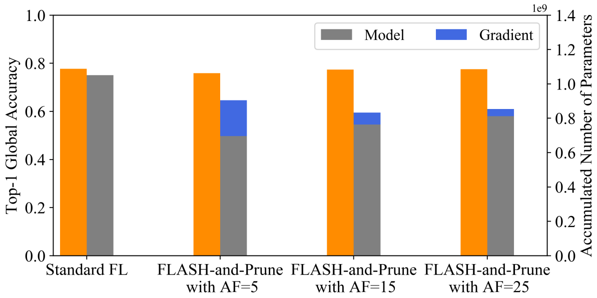


Fig. 7: Top-1 global accuracy and accumulated number of parameters in the uplink (including both model and gradient) for standard FL and three FLASH-and-Prune schemes with different AFs.

cost of transmitting all the data to a central unit. Finally, we present the performance of standard FL, where the vehicles collaboratively learn a shared model by sharing local model updates. In Fig. 6a, the star, dot, and triangle markers show the standard FL accuracy at iterations 10, 40, and 150.

Observation 1. *We observe that local learning fails to achieve competitive performance when exposed to the global test dataset. Moreover, in order to achieve 77.70% top-1 accuracy, the centralized learning requires data from around 8 vehicles, while the standard FL architecture achieves the same accuracy without data sharing and with only 150 iterations of aggregation.*

6.2 Benefit of Fusion

In Fig. 6b, we consider the standard FL architecture and compare the top-1 global accuracy when using only LiDAR data versus fusion technique described in Sec. 4.2. We choose LiDAR as it demonstrates the maximum accuracy of 80.37% in centralized learning for unimodal models, compared to 68.75% and 29.81% for image and GPS, respectively. We observe that while both methods experience improvement over federated iterations, the fusion architecture explicitly outperforms the LiDAR-only data with a maximum improvement of 34.37% in the top-1 global accuracy.

Observation 2. *We observe that taking advantage of fusion architecture boost the learning rate compared to the most successful sensor modality, i.e. LiDAR.*

6.3 FLASH-and-Prune and Global Inference

We study the performance of the FLASH-and-Prune scheme described in Sec. 4.4 with respect to different design parameters such as adjustment frequency (AF) and initial pruning. Moreover, we provide comparisons with two other state-of-the-art pruning strategies.

6.3.1 *Effect of Adjustment Frequency*

The original fusion model described in Sec. 5.3 (See Fig. 5) includes 6,957,992 trainable parameters. In a standard FL architecture, all model parameters must be transmitted in each federated iteration. However, with FLASH-and-Prune scheme, the number of model parameters is optimized at each adjustment iteration that happens at the interval of the AF between federated iterations. In an adjustment iteration, both the model weights and gradients are considered as model parameters to be exchanged and a pruning step is applied on the global model. In Fig. 7, we compare the accumulated uplink overhead for four experimental settings: (i) standard FL, (ii) FLASH-and-Prune with an AF of 5, (iii) FLASH-and-Prune with an AF of 15, iv) FLASH-and-Prune with an AF of 25. For all FLASH-and-Prune experiments, our analysis includes the overhead associated with exchanging both the local model weights and gradient. We observe that all three pruning experiments achieve the same accuracy as the standard FL; thus, yielding the same performance. On the other hand, all three pruning methods outperform standard FL in overhead. In particular, FLASH-and-Prune with AFs of 5, 15 and 25 exhibit 13.91%, 20.73% and 18.76% less overhead than standard FL, respectively. Interestingly, we observe that although pruning at every fifth iteration decreases the overall model size, the overhead associated with sending the gradients results in a larger overall overhead in comparison with two other AFs. On the other hand, pruning with an AF of 25 has less overhead with respect to gradient; however, the overall model compression rate is also less. This implies that the AFs play a dominant role in the proposed FLASH-and-Prune scheme.

6.3.2 *Effect of Initial Pruning*

We argue that the initial pruning step at a selected client further decreases the exchange overhead. Fig. 8a denotes the top-1 global accuracy of standard FL in comparison with FLASH-and-Prune with and without initial pruning for an AF of 15. We observe that all three competing methods perform closely with respect to the ultimate accuracy and learning rate over federated iterations. The adjustments in the model size are shown in Fig. 8b, where standard FL has a constant number of model parameters over federated iterations. On the other hand, FLASH-and-Prune without initial pruning gradually adjusts the model size starting from the original model while the case with initial pruning experiences a drop in model size in the first iteration and gradually adapts the model later. This results in much less overhead while promising a close performance in the accuracy. In Tab. 2, we compare FLASH-and-Prune with and without initial pruning against standard FL with respect to accuracy and compression rate in the accumulated uplink overhead. We conclude that initial pruning affects the accuracy by 1.03-1.19%, but the accumulated overhead (including the gradient) is 9.19-17.92% less than the case

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

Case	Without Initial Pruning		With Initial Pruning	
	Accuracy	Compression w.r.t Standard FL	Accuracy	Compression w.r.t Standard FL
Standard FL	77.70%	-	-	-
FLASH-and-Prune AF=15	77.36%	20.73%	76.54%	29.92%
FLASH-and-Prune AF=25	77.57%	18.76%	74.68%	36.68%

TABLE 2: Effect of initial pruning on FLASH-and-Prune with respect to the accumulated number of parameters in the uplink for AFs of 15 and 25.

without initial pruning. Our experiments reveal that the top-1 accuracy does not improve significantly after 150 iterations (only 6.78% improvement with 250 more iterations). Thus, we compare the results for 150 iterations only.

6.3.3 Effect of Number of Participating Users

In Fig. 8c, we compare the performance of FLASH-and-Prune when different number of users participate in federated iterations. In this experiment, we set the adjustment frequency to be 15 and include the initial pruning step, which results in the optimum performance according to Tab. 2. We consider a scenario where $2, 4, 6, 8, 10$ users are participating in federated learning. These model updates are stored in the buffer and the MEC performs aggregation once it receives the updates from a few vehicles. From this experiment, we observe that having more users improves the accuracy, as expected. In particular, the maximum achieved accuracy for $2, 4, 6, 8$ and 10 users are 33.86%, 48.09%, 62.64%, 72.25%, and 85.15%, respectively. Note that this experiment corresponds to the worst case scenario, where the BS does not receive enough model updates in the uplink. However, since the training phase is asynchronous in FLASH-and-Prune, the BS can wait for receiving a few model updates from the vehicles before computing the global model.

6.3.4 Comparison with State-of-the-art Pruning Methods

We compare the performance of the proposed FLASH-and-Prune method against two other pruning strategies methods on the FLASH dataset. First, we consider an iterative pruning scheme [15], one of the most popular pruning strategies, where the model size is reduced over multiple iterations and retrained after each pruning step. Second, we compare against Single-Shot Network Pruning based on Connection Sensitivity (SNIP) [16] that identifies structurally important weights in the network before starting the federated iterations. For our pruning method, we set the AF to be 15 and include the initial pruning step. Fig. 9 compares the performance of proposed FLASH-and-Prune with two aforementioned competing methods with respect to top-1 global accuracy and model size over federated iterations, while targeting the same model density. We observe that the proposed FLASH-and-Prune method outperforms both iterative pruning and SNIP by 28.08% and 30.94% in accuracy at iteration 150. In particular, the SNIP drastically decreases the model size before starting the federated iterations; however, it fails to achieve the same accuracy as FLASH-and-Prune (30.94% drop in top-1 global accuracy). On the other hand, the iterative pruning gradually removes the weights at each federated iteration, yet it cannot compete with the compression rate achieved by FLASH-and-Prune. Moreover, it results in a 28.08% drop in top-1 global accuracy.

Observation 3. *The FLASH-and-Prune without initial pruning decrease the overhead by 13.91-20.73% compared to standard FL. It offers 9.19-17.92% extra improvement with initial pruning with neglectable drop in accuracy (∼ 1%). In FLASH-and-Prune, the accuracy increases with the number of participating vehicles. It also shows superiority over benchmark pruning methods.*

Methods	Modalities	Architecture	Top-1 Acc. (%)	Dataset	Evaluation Type
Klautau *et al.* [22]	LiDAR	Centralized	30.5 ± 1	Raymobtime [42]	Synthetic
Dias *et al.* [23]	LiDAR	Centralized	20.5 ± 1	Raymobtime [42]	Synthetic
Standard FL	GPS, Image, LiDAR	Distributed	**77.70**	FLASH [39]	Testbed
FLASH	GPS, Image, LiDAR	Distributed	**59.72**	FLASH [39]	Testbed
FLASH-and-Prune	GPS, Image, LiDAR	Distributed	**77.57**	FLASH [39]	Testbed

TABLE 3: Comparing FLASH-and-Prune with the state-of-the-art techniques which use non-RF data for sector selection.

6.4 FLASH Architectures vs State-of-the-art

In Tab. 3, we benchmark the performance of our proposed FLASH-and-Prune architecture against the state-of-the-art DL-based approaches by Klautau *et al.* [22] and Dias *et al.* [23]. Both of these techniques use centralized learning with only LiDAR sensors at the vehicle while considering both LOS and NLOS situations on synthetically-generated Raymobtime dataset [42]. Moreover, we compare against FLASH [40] where different branches of the models are selected through a multimodal orchestrator. We limit the comparison study to the above techniques, as the other state-of-the-art techniques differ from ours with respect to various aspects, such as: (a) different evaluation metrics [20], [43], [44]; (b) consideration of LOS-only scenarios while using camera sensors [45]; and (c) inclusion the RF inputs [21].

Observation 4. *From Tab. 3, we observe that FLASH-and-Prune outperform the state-of-the-art by 17-57% in top-1 accuracy and maintains close competing accuracy compared to an standard FL architecture.*

6.5 Accuracy and Overhead Trade-off

Both the centralized and federated learning based methods impose some communication overhead in the control channel for *model initialization*. We observe a trade-off between overhead and accuracy over all five competing methods, presented in Tab. 4. This analysis includes the average overhead per iteration. Moreover, we use *float16* data type to compute the model sizes.

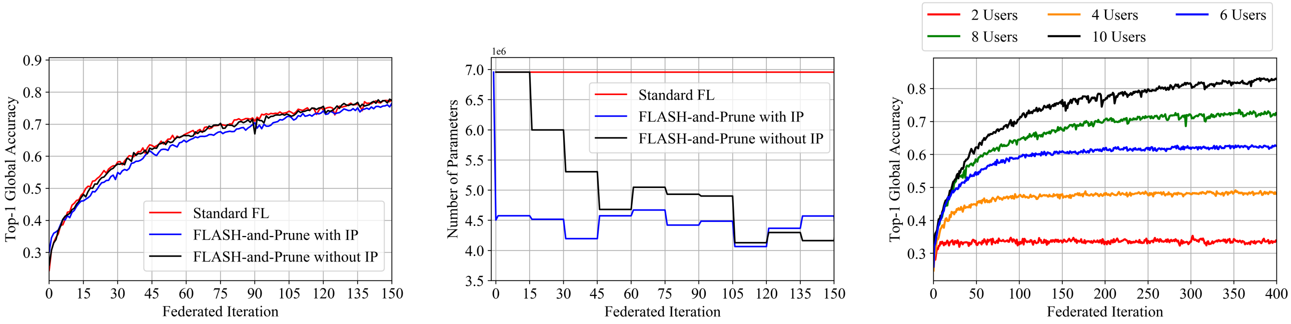
Local Learning. Though the local learning approach does not require any data/model sharing, it provides up to only 36.78% top-1 accuracy.

Centralized Learning. The centralized learning approach provides 87.31% accuracy, but it comes with a large communication cost of transmitting the data of all vehicles (∼2.5 GB per iteration) to the cloud, as well as privacy concerns. Moreover, the trained model must be transmitted to all clients after the training is completed at the MEC.

Standard FL. This approach reduces the communication cost while preserving 77.70% accuracy by only sharing the local models. This imposes 13.35MB overhead in the uplink and downlink over federated iterations. Hence, we

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

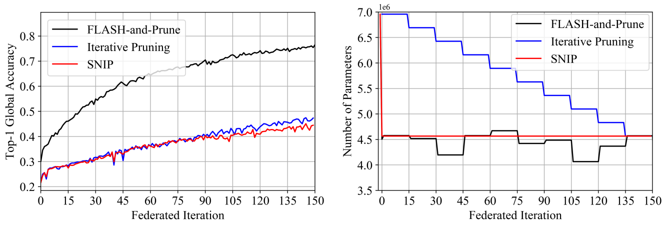


(a)

(b)

(c)

Fig. 8: (a) Top-1 global accuracy and (b) fluctuations in the number of parameters with standard FL versus FLASH-and-Prune with and without initial pruning (IP) for an AF of 15. (c) Top-1 global accuracy over federated iteration for different number of users. The top-1 accuracy increases with the number of users. However, since the training phase is offline in FLASH-and-Prune, the BS can wait to receive the model updated from different vehicles.



(a)

(b)

Fig. 9: Comparing the performance of the proposed FLASH-and-Prune framework against iterative pruning and SNIP with respect to (a) top-1 global accuracy and (b) model size over federated iterations.

Methodology	Acc (%)	Overhead		
		Data Sharing	Model Sharing	
			Uplink ($\widetilde{o_{ul}}$)	Downlink ($\widetilde{o_{dl}}$)
Local Learning	36.78	-	-	-
Centralized Learning	87.31	2.5GB	-	13.35MB
Standard FL	77.70	-	13.35MB	13.35MB
FLASH	59.72	-	13.35MB	3.31MB
FLASH-and-Prune	77.36	-	9.45MB	8.56MB

TABLE 4: Comparing the performance of the five data-driven competing methods with respect to accuracy and average model initialization overhead per iteration. All accuracies are reported on the global test set.

conclude that standard FL provides a 40.92% improvement in accuracy over local learning and 97.93% improvement in overhead over centralized learning.

FLASH. In the FLASH framework, one out of four branches is sent back to the vehicles at each aggregation iteration. Retrieving the global model in the downlink with unbiased policy $\mathcal{P}_{\text{Unbiased}}$ which uniformly selects a branch for aggregation requires sending 13.35MB of data. Considering both uplink and downlink, FLASH imposes 37.50% less overhead than standard FL architecture; however; it comes with the cost of 17.98% drop in the top-1 global accuracy.

FLASH-and-Prune. Finally, FLASH-and-Prune observes only 0.34% (<1%) drop in accuracy with strategic weight pruning according to Eq. (15). It also reduces the overhead by 29.25% and 35.89% in uplink and downlink, respectively, compared to standard FL architecture.

Observation 5. *We observe that FLASH-and-Prune outperforms (a) local learning by 40.92% in accuracy, (b) centralized learning by 90.55% in overhead, (c) standard FL by 34.94% in overhead, and (d) FLASH by 17.98% in accuracy. We conclude that FLASH-and-Prune is the most successful framework out of the five with the lowest overhead and comparable prediction accuracy.*

6.6 Discussions and Requirements

Despite demonstrates promising results, we discuss implications of some of our assumptions and limitations that may impact performance in practical scenarios. In this section, we discuss these limitations and identify possible future directions to address them.

- **Sector Selection Speed:** At the inference phase, FLASH-and-Prune predicts the optimum sector ID from the multimodal sensor data by following four steps: (a) *Data acquisition*: given the high-sampling rates of COTS sensors, we assume that sensor data is acquired almost instantaneously. Moreover, the LiDAR preprocessing step described in Sec. 4.1 has a negligible latency that can be further reduced by exploiting parallel processing; (b) *Model inference*: we pass a test sample 100 times over the DL model and calculate the average inference delay of 0.6 ms; (c) *Sector sharing*: an integer varying between 0-31 and 61-63, representing the selected sector ID is sent back to the BS. Even though the sector ID is only an integer, sharing it with the BS requires sending an entire frame. The slot duration is equal to 1ms in 5G standard with numerology 0 [46]. Moreover, considering the baseband Tx and Rx processing delays the overall overhead is 3× of the slot time. As a result, the end-to-end sector selection time with FLASH-and-Prune ranges between 1.6ms-3.6ms. On the other hand, the exhaustive search proposed by 802.11ad and 5G-NR take 1.27 ms (experimentally measured in [5]) and 2.65 ms [32], [47], respectively. As a result, the beam selection time with FLASH-and-Prune may exceed the exhaustive search for small codebooks. The FLASH-and-Prune approach will need to be thoroughly revised to enable competitive performance for small codebooks. A possible solution is using a proactive prediction mechanism [48] to obtain the optimal sector prior to the arrival of the vehicle at a certain location or incorporating customized control channels that provide overhead of $< 1ms$ to outperform the current exhaustive

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.

© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

search based method.

- **FLASH-and-Prune for Large Codebooks:** In this paper, we used the FLASH dataset for validation, as the only real world dataset for the mmWave vehicular networks. This dataset is collected using Talon AD7200 radios. In order to collect the data, the firmware of this radio is hacked to access the SNR and received signal strength (RSSI) for different sectors. This process enables accessing to the profiles for 34 sectors. However, the entire codebook in this radio has 64 sectors. Another example is the National Instruments mmWave radio [49] that has a codebook with 16 sectors, which also enables beamforming at both Tx and Rx for maximum efficiency; thus, a total of 256 sector combinations are possible. Taken together, the COTS mmWave radios with higher resolution have more sectors in their codebooks. In this case, the sector selection overhead with these two radios and 5G-NR standard are 4.992 ms and 64.914 ms, respectively. On the other hand, the overhead of FLASH-and-Prune with larger codebooks does not increase significantly (nor exponential) at the inference. Thus, we conclude that while FLASH-and-Prune results in higher overhead in small codebooks compared to exhaustive search based methods, the significance of reduction in the sector selection time is more pronounced with larger codebooks [7].
- **Asynchronous Model Update:** In a synchronous federated learning setting, the federated serve sends the model update requests to the users. If a user decides to participate in FL, it will share its local model update to the BS and the participating users send their local model update in the uplink, simultaneously. This may affect the experience of the other regular users if the entire channel is designated to the FLASH-and-Prune, or it may further increase their overhead if there is contention between the FLASH-and-Prune and other regular users [50]. Thus, we opt for buffered asynchronous model aggregation in our design, where the vehicles share the model updates when they are stationary (for example at the end of the trips where they could simply use their home WiFi). The federated server in our design collects and stores the model updates from a few vehicles in a buffer, and performs the aggregation to obtain the global model. Similarly, the vehicles can download the most updated global model at the start of each trip or at the start of a day. In an asynchronous setting, the federated server buffers the model updates, and generates the global model after receiving the model updates from a few vehicles. Thus, the experiments results in this paper, which are generated based on 10 participating users, hold, as in an asynchronous setting the federated server generates the global model whenever the model updates from 10 vehicles are received. Nevertheless, for the completeness of our analysis, we provided results for cases, where less than 10 users participate in federated iterations in Fig. 8c.

7 Conclusions

We make a case for using multiple sensor modalities to aid in mmWave beamforming, as opposed to using only RF-based approaches. FLASH-and-Prune incorporates DL based multimodal data fusion using architectures where training and dissemination in real-world vehicular networks is achieved using a federated learning architecture. FLASH-and-Prune also employs a pruning algorithm that is customized for distributed federated learning architecture and reduces the model parameter exchange overhead by 29.25% and 35.89% in uplink and downlink, respectively, while maintaining the accuracy. The FLASH dataset is already available at [39] and the codebase for FLASH-and-Prune will be released in the same repository upon the acceptance of this article.

References

[1] J. Choi, V. Va, N. González-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, “Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[2] I. Rasheed, F. Hu, Y. Hong, and B. Balasubramanian, “Intelligent Vehicle Network Routing With Adaptive 3D Beam Alignment for mmWave 5G-Based V2X Communications,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.

[3] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, “Millimeter-Wave Beamforming as an Enabling Technology for 5G Cellular Communications: Theoretical Feasibility and Prototype Results,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 106–113, 2014.

[4] A. Vizziello, P. Savazzi, and K. R. Chowdhury, “A kalman based hybrid precoding for multi-user millimeter wave mimo systems,” *IEEE Access*, vol. 6, pp. 55 712–55 722, 2018.

[5] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, “Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11ad Devices,” *International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2017.

[6] N. González-Prelcic, A. Ali, V. Va, and R. W. Heath, “Millimeter-Wave Communication with Out-of-Band Information,” *IEEE Communications Magazine*, vol. 55, no. 12, pp. 140–146, 2017.

[7] B. Salehi, U. Demir, D. Roy, S. Pradhan, J. Dy, S. Ioannidis, and K. Chowdhury, “Multiverse at the edge: Interacting real world and digital twins for wireless beamforming,” *arXiv preprint arXiv:2305.10350*, 2023.

[8] D. Roy, B. Salehi, S. Banou, S. Mohanti, G. Reus-Muns, M. Belgiovine, P. Ganesh, C. Dick, and K. Chowdhury, “Going beyond rf: A survey on how ai-enabled multimodal beamforming will shape the nextg standard,” *Computer Networks*, vol. 228, p. 109729, 2023.

[9] J. Gu, B. Salehi, D. Roy, and K. R. Chowdhury, “Multimodality in mmwave mimo beam selection using deep learning: Datasets and challenges,” *IEEE Communications Magazine*, vol. 60, no. 11, pp. 36–41, 2022.

[10] J. Gu, B. Salehi, S. Pimple, D. Roy, and K. R. Chowdhury, “Tune: Transfer learning in unseen environments for v2x mmwave beam selection,” in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 1658–1663.

[11] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, “A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking,” in *IEEE intelligent transportation systems conference*. IEEE, 2007, pp. 1044–1049.

[12] A. Festag, “Standards for Vehicular Communication—from IEEE 802.11 p to 5G,” *Elektrotech. Inftech.*, vol. 132, no. 7, pp. 409–416, 2015.

[13] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 54, 2017, pp. 1273–1282.

[14] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, “Rethinking the value of network pruning,” *arXiv preprint arXiv:1810.05270*, 2018.

[15] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” *Advances in neural information processing systems*, vol. 28, 2015.

[16] N. Lee, T. Ajanthan, and P. H. Torr, “Snip: Single-shot network pruning based on connection sensitivity,” *arXiv preprint arXiv:1810.02340*, 2018.

[17] J. Palacios, D. Steinmetzer, A. Loch, M. Hollick, and J. Widmer, “Adaptive Codebook Optimization for Beam Training on Off-the-Shelf IEEE 802.11ad Devices,” *International Conference on Mobile Computing and Networking (MobiCom)*, 2018.

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

[18] S. K. Saha, H. Assasa, A. Loch, N. M. Prakash, R. Shyamsunder, S. Aggarwal, D. Steinmetzer, D. Koutsonikolas, J. Widmer, and M. Hollick, "Fast and Infuriating: Performance and Pitfalls of 60 GHz WLANS Based on Consumer-Grade Hardware," *IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2018.

[19] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, "Wifi-Assisted 60 GHz Wireless Networks," in *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2017, pp. 28–41.

[20] 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.

[21] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter Wave Base Stations with Cameras: Vision-Aided Beam and Blockage Prediction," in *IEEE 91st Vehicular Technology Conference (VTC2020)*, 2020, pp. 1–5.

[22] 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.

[23] M. Dias, A. Klautau, N. González-Prelcic, and R. W. Heath, "Position and LIDAR-aided mmWave Beam Selection using Deep Learning," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[24] G. Reus-Muns, B. Salehi, D. Roy, T. Jian, Z. Wang, J. Dy, S. Ioannidis, and K. Chowdhury, "Deep learning on visual and location data for v2i mmwave beamforming," in *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. IEEE, 2021, pp. 559–566.

[25] X. Yao, T. Huang, C. Wu, R. Zhang, and L. Sun, "Towards Faster and Better Federated Learning: A Feature Fusion Approach," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 175–179.

[26] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning design," *arXiv preprint arXiv:2012.08336*, 2020.

[27] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2457–2471, 2020.

[28] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 1935–1949, 2020.

[29] W. Xu, W. Fang, Y. Ding, M. Zou, and N. Xiong, "Accelerating federated learning for iot in big data analytics with pruning, quantization and selective updating," *IEEE Access*, vol. 9, pp. 38 457–38 466, 2021.

[30] Y. Jiang, S. Wang, V. Valls, B. J. Ko, W.-H. Lee, K. K. Leung, and L. Tassiulas, "Model pruning enables efficient federated learning on edge devices," *arXiv preprint arXiv:1909.12326*, 2019.

[31] T. Nitsche, A. B. Flores, E. W. Knightly, and J. Widmer, "Steering with Eyes Closed: mm-Wave Beam Steering without In-band Measurement," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 2416–2424.

[32] C. N. Barati, S. Dutta, S. Rangan, and A. Sabharwal, "Energy and Latency of Beamforming Architectures for Initial Access in mmWave Wireless Networks," *Journal of the Indian Institute of Science*, pp. 1–22, 2020.

[33] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, and D. Huba, "Federated learning with buffered asynchronous aggregation," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 3581–3607.

[34] Z. Wang, Z. Zhang, Y. Tian, Q. Yang, H. Shan, W. Wang, and T. Q. Quek, "Asynchronous federated learning over wireless communication networks," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 6961–6978, 2022.

[35] L. Bonati, M. Polese, S. D'Oro, S. Basagni, and T. Melodia, "Open, programmable, and virtualized 5g networks: State-of-the-art and the road ahead," *Computer Networks*, vol. 182, p. 107516, 2020.

[36] T. Jian, D. Roy, B. Salehi, N. Soltani, K. Chowdhury, and S. Ioannidis, "Communication-aware dnn pruning," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications*. IEEE, 2023, pp. 1–10.

[37] B. Salehi, D. Roy, T. Jian, C. Dick, S. Ioannidis, and K. Chowdhury, "Omni-cnn: A modality-agnostic neural network for mmwave beam selection," *IEEE Transactions on Vehicular Technology*, 2024.

[38] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[39] "FLASH Dataset." [Online]. Available: https://genesys-lab.org/multimodal-fusion-nextg-v2x-communications

[40] 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.

[41] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[42] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-selection Using Deep Learning," in *2018 Information Theory and Applications Workshop (ITA)*. IEEE, 2018, pp. 1–9.

[43] V. Va, T. Shimizu, G. Bansal, and R. W. Heath, "Beam Design for Beam Switching Based Millimeter Wave Vehicle-to-Infrastructure Communications," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[44] J. C. Aviles and A. Kouki, "Position-aided mm-Wave Beam Training Under NLOS Conditions," *IEEE Access*, vol. 4, pp. 8703–8714, 2016.

[45] W. Xu, F. Gao, S. Jin, and A. Alkhateeb, "3D Scene-Based Beam Selection for mmWave Communications," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1850–1854, 2020.

[46] "3GPP TS 38.214 5G NR Physical Layer procedures for data," http://www.etsi.org/standards-search.

[47] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 2018.

[48] B. Lin, F. Gao, Y. Zhang, C. Pan, and G. Liu, "Multi-camera view based proactive bs selection and beam switching for v2x," *arXiv preprint arXiv:2207.05299*, 2022.

[49] "Ni radio," https://www.ni.com/en/shop/wireless-design-test/what-is-mmwave-transceiver-system.html.

[50] C. Xu, Y. Qu, Y. Xiang, and L. Gao, "Asynchronous federated learning on heterogeneous devices: A survey," *Computer Science Review*, vol. 50, p. 100595, 2023.

Batool Salehi is currently pursuing a Ph.D. degree in computer engineering at Northeastern University under the supervision of Prof. K. Chowdhury. Her current research focuses on mmWave beamforming, Internet of Things, and the application of machine learning in the domain of wireless communication.

Debashri Roy received her MS (2018) and PhD (2020) degrees in Computer Science from University of Central Florida, USA. She is currently associate research scientist at Northeastern University. Her research interests are in the areas of AI/ML enabled technologies in wireless communication, multimodal data fusion, nextG networks, and networked systems.

Jerry Gu is pursuing a Ph.D. degree in Computer Engineering at Northeastern University, where he received his M.S. (2021) in Computer Engineering. His current research focuses on the use of ML in wireless communications, including multimodal data fusion and RF fingerprinting.

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Mobile Computing. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TMC.2024.3401046

Chris Dick is a wireless architect at NVIDIA and the technical lead for the application of AI and machine learning to 5G and 6G wireless. In his 24 years working in signal processing and communications he has delivered silicon and software products for 3G, 4G, and 5G baseband DSP and Docsis 3.1 cable access. He has performed research and delivered products for digital frontend (DFE) technology for cellular systems.

Kaushik Chowdhury is a Professor at Northeastern University, Boston, MA. He is presently a co-director of the Platforms for Advanced Wireless Research (PAWR) project office. His current research interests involve systems aspects of networked robotics, machine learning for agile spectrum sensing/access, wireless energy transfer, and large-scale experimental deployment of emerging wireless technologies.

Authorized licensed use limited to: Northeastern University. Downloaded on May 18,2024 at 14:12:02 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.