FLASH: Federated Learning for Automated Selection of High-band mmWave Sectors

Batool Salehi, Jerry Gu, Debashri Roy, and Kaushik Chowdhury
Department of Electrical and Computer Engineering
Northeastern University, Boston, MA, USA
E-mail: {bsalehihikouei,jgu1,droy, krc}@ece.neu.edu

***Abstract*—Fast sector-steering in the mmWave band for vehicular mobility scenarios remains an open challenge. This is 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. The contributions in this paper are threefold: First, a multimodal deep learning architecture is proposed that fuses the inputs from these data sources and locally predicts the sectors for best alignment at a vehicle. Second, it studies the impact of missing data (e.g., missing LiDAR/images) during inference, which is possible due to unreliable control channels or hardware malfunction. Third, it describes the first-of-its-kind multimodal federated learning framework that combines model weights from multiple vehicles and then disseminates the final fusion architecture back to them, thus incorporating private sharing of information and reducing their individual training times. We validate the proposed architectures on a live dataset collected from an autonomous car equipped with multiple sensors (GPS, LiDAR, and camera) and roof-mounted Talon AD7200 60GHz mmWave radios. We observe 52.75% decrease in sector selection time than 802.11ad standard while maintaining 89.32% throughput with the globally optimal solution.**

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

I. 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 available in the 57-72 GHz frequency range.

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. An exhaustive search of all possible configurations results in a large overhead. Hence, current standards, such as IEEE 802.11ad, prescribe a set of predefined patterns, referred to as *sectors* [4], with a deterministic sweeping algorithm that selects the optimal sector with the strongest mmWave link between transmitter (Tx) and receiver (Rx). The 802.11ad standard, in particular, proposes 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.

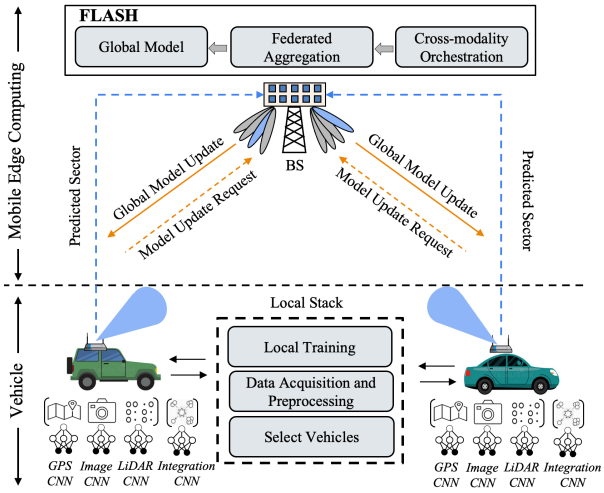


Fig. 1: The schematic of proposed FLASH framework for mmWave vehicular networks, where each vehicle is equipped with GPS, LiDAR and camera sensors.

A. 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, Rx, and potential obstacles play an important role in the sector selection process. 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. These sensors are present in autonomous vehicles to aid in driving [5] but can also be re-purposed to optimize communication links. Furthermore, with regard to mapping, using multiple modalities increases resilience, wherein missing information from a

particular sensor type can be compensated by utilizing data from the others, with graceful degradation of performance.

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 propose a deep learning (DL) framework that uses non-RF sensor data to select the best sector to probe *without* attempting an exhaustive search. Once the best sector is determined, the BS starts the multi-Gbps downlink transmission to the vehicle, instantaneously. The proposed DL-based inference engine in each vehicle is resilient to missing data; even if some data modalities are missing at any given time, the engine is capable of generating remarkably accurate predictions of the best sector. We note that multiple sensors are now included as standard installations both in modern cars and roadside infrastructures [5]: LiDAR and camera sensors are already indispensable parts of modern vehicles, used for driving corrections and collision avoidance [6]; GPS data is regularly collected and transmitted as part of basic safety messages in V2X applications [7].

B. 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 [8]. 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 global shared model at the MEC, and then disseminated back to the vehicles for local inference; this is also shown in Fig. 1. 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 centralized MEC. Moreover, we use a sub-6 GHz control channel to relay model weight updates.

Note that using a multitude of sensor modalities improves the prediction performance by providing a comprehensive representation of the environment. Moreover, it gives the flexibility to adjust the contribution of each modality to the federated aggregation iterations according to their performance optimality on a case-by-case basis. For example, GPS works reliably in LOS-dominant environments, such as open freeways, while LiDAR, giving a 3-D representation, is more effective in an NLOS-heavy environment such as an urban canyon, where buildings flank the road on both sides. Besides, LiDAR and camera performances are prone to errors in the presence of strong sunlight reflections [9] and low light conditions, respectively. Hence, a selective approach may improve the overall performance by being biased towards situationally-favored modalities.

C. Our Contributions

Our main contributions are as follows:

- We design robust DL 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 show that adding more viewpoints in the training data enhances the performance of sector selection and analyze the resulting control overhead.
- We propose FLASH, a multimodal FL framework, where local DL model weights are globally optimized by fusing them at the MEC. So far, the state-of-the-art in FL has focused on unimodal data, which suggests that FLASH may be suitable for other generalized problems involving multiple data types (beyond mmWave beamforming).
- We describe a multimodal data adaptation technique that is executed in the individual vehicles, making FLASH resilient to missing sensor information. We observe 67.59% top-1 accuracy even when all sensors are missing for 10 consecutive samples.
- We rigorously analyze the end-to-end latency of FLASH and compare it with IEEE 802.11ad standard and demonstrate that sector selection time decreases by 52.75% on average while maintaining 89.32% of the throughput. Due to lack of access to programmable cellular 5G mmWave BS and clients, we use two 802.11ad-enabled mmWave Talon routers to evaluate FLASH on real-world scenarios. Without loss of generality, FLASH can be applied to other bands and wireless standards.
- We publish the first (to the best of our knowledge) dataset collected by an autonomous vehicle mounted with multimodal sensors and mmWave radios for community use in [10]. The dataset includes comprehensive settings of LOS and NLOS scenarios for the urban canyon region.

II. Related Works

We survey the most relevant articles that use auxiliary information to reduce the sector selection overhead. Steinmetzer *et al.* [4] propose a compressive path tracking algorithm where the measurements on a random subset of sectors are used to estimate the optimum sector. In [11], 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.* [12] 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.* [13] 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.* [14] use the location of all nearby vehicles, including the target Rx, as the input for their sector inference algorithm, while Alrabeiah *et al.* [15] combine both camera images and a recorded sequence of previous sectors to model dynamic mmWave communication in outdoor scenarios. Klautau *et al.* [16] and Dias *et al.* [17] propose to reduce the sector search space using GPS and LiDAR sensors in vehicular settings. On the other hand, Muns *et al.* [18] use GPS and camera images to speed up the beam 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. Although FL provides frameworks to overcome the security risks with a reduced overhead [8], recent works attempt to reduce such overheads further [19].

III. Flash System Architecture

In this section, we first review classical sector initialization methods and then propose a distributed system architecture that uses non-RF data from multiple sensors.

A. Traditional Beam Initialization

The IEEE 802.11ad standard sector initialization steps consist of two stages that starts with a mandatory sector level sweep (SLS) and follows with an optional beam refinement process (BRP). During SLS, 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-omni-directional antenna setting. This process is then repeated with the *initiator* and *responder* roles reversed. In the SLS phase, a complete frame must be transmitted at each sector in the lowest PHY rate, incurring a time cost of $\sim$1.27ms for only 34 sectors [4]. The BRP is used to fine-tune the sectors detected in the SLS phase. As it uses only one frame, the BRP imposes much less overhead. Hence, we focus on SLS phase, as it generates the largest overhead.

B. Problem Statement

Consider a Tx and Rx pair equipped with phased antenna arrays with a predefined codebook defined by $C_{Tx} = \{t_1, \ldots, t_M\}$, $C_{Rx} = \{r_1, \ldots, r_N\}$ consisting of M and N elements, respectively. A total of $M + N$ probe frames must be transmitted to complete the SLS 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:

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

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

C. FLASH with Multimodal Learning

From Sec. III-B, 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. Thus, we propose a learning framework to exploit multiple sensor measurements that can directly estimate the best sector t^* in one shot and then immediately start the transmission. Our proposed solution consists of the following four components:

- **Data Acquisition and Preprocessing:** The collected sensor data first passes through the preprocessing phase. For LiDAR, 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. IV-A. We also define a new coordinate system to effectively merge the decimal degree GPS and metric LiDAR measurements.
- **Local Training at the Semi-autonomous Edge:** Given preprocessed multimodal sensor data, we design a fusion architecture that is trained over local data (i.e., the data available at a given vehicle or each semi-autonomous edge). We design a novel fusion network that combines all the modalities for the local training; refer to Sec. IV-B.
- **Multimodal Federated Training:** Given the locally trained models for each unimodal and fusion network, we propose a multimodal FL-based architecture as a global optimization technique; see Sec. IV-C
- **Resilient Inference:** Finally, we include measures to make the inference through the trained and optimized fusion architecture adaptive to the unavailable sensor data at the edge; refer to Sec. IV-D.

IV. FLASH Framework Design

A. Data Acquisition and Preprocessing

Multimodal data from GPS, camera, and LiDAR sensors is collected and passed through preprocessing steps as follows.

1) LiDAR Preprocessing: 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 [17]. 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.

2) GPS Coordinate System: The raw GPS coordinates recorded at the vehicle are in Decimal Degree; however, the LiDAR data are in meters. We consider a fixed-origin and calculate absolute distances from that origin to define a Cartesian coordinate system [20]. In regard to the LiDAR system, points are measured with respect to the sensor location, i.e., the vehicle position. Thus, we adjust the LiDAR point clouds by the difference between two origins pertaining to the GPS and LiDAR coordinate systems.

Algorithm 1: Multimodal federated training

Input: Initial parameters $\theta_\nu^{\mathtt{FN}(0)} = \theta^{\mathtt{FN}(0)}\ \forall \nu \in V$ *(at vehicles)*
$\mathcal{P} = \{\alpha, \beta, \gamma, \delta\}$,where $\alpha + \beta + \gamma + \delta = 1$ *(at MEC)*
Output: Trained global model weights $\theta_\nu^{\mathcal{B}(i)}$
for *each* $i = 1 \dots \mathcal{N}$ **do**
  $\theta_\nu^{\mathtt{FN}(i)}$ = local training for ξ ephs on $\theta_\nu^{\mathtt{FN}(i-1)}$ *(at vehicles)*
  Each participating vehicle ν shares $\theta_\nu^{\mathtt{FN}(i)}$ to MEC
  Assign four branches $\mathcal{B}_{\mathtt{C}}^{(i)}$, $\mathcal{B}_{\mathtt{I}}^{(i)}$, $\mathcal{B}_{\mathtt{L}}^{(i)}$, $\mathcal{B}_{\mathtt{IN}}^{(i)}$ within $\theta_\nu^{\mathtt{FN}(i)}$
  $\mathcal{B}(i) = \mathbf{P}_{\mathcal{P}}(\mathcal{B}_{\mathtt{C}}^{(i)}, \mathcal{B}_{\mathtt{I}}^{(i)}, \mathcal{B}_{\mathtt{L}}^{(i)}, \mathcal{B}_{\mathtt{IN}}^{(i)})$ *(at MEC)*
  MEC computes $\theta^{\mathcal{B}(i)} = \frac{1}{|V|}\sum_{v=1}^{V} \theta_\nu^{\mathcal{B}(i)}$
  MEC distributes $\theta^{\mathcal{B}(i)}$ such that $\theta_\nu^{\mathcal{B}(i)} = \theta^{\mathcal{B}(i)}\ \forall \nu \in V$
end

B. 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, latitude and longitude. The label matrix $Y_\nu \in \{0,1\}^{N_t \times M}$ represents the one-hot encoding of 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 can use a DNN-based unimodal network to extract discriminative features from the input and infer the optimum sector. Each unimodal network makes a probabilistic prediction of the best sector through softmax layer σ defined as:

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

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

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

where $f_{\mathtt{C}}^{\nu}(.)$, $f_{\mathtt{I}}^{\nu}(.)$, $f_{\mathtt{L}}^{\nu}(.)$ denotes the unimodal network for each vehicle ν parameterized by $\theta_\nu^{\mathtt{C}}$, $\theta_\nu^{\mathtt{I}}$, $\theta_\nu^{\mathtt{L}}$. 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 $f_{\mathtt{IN}}^{\nu}(.)$ parameterized by $\theta_\nu^{\mathtt{IN}}$, as presented in Fig. 1. Formally,

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

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

where $f_{\mathtt{FN}}^{\nu}(.)$ is the fusion model parameterized by $\theta_\nu^{\mathtt{FN}}$. Finally, the prediction happens at the output of fusion network through the computation of $\mathbf{s} = \sigma(\mathbf{u}_{\mathtt{FN}}^{\nu})$. The sector that has highest score is chosen as the predicted sector. We refer to each component of the *fusion* network as *branches*; i.e., (a) GPS branch ($\mathcal{B}_{\mathtt{C}}$), $f_{\theta^{\mathtt{C}}}^{\nu}(.)$; (b) image branch ($\mathcal{B}_{\mathtt{I}}$), $f_{\theta^{\mathtt{I}}}^{\nu}(.)$; (c) LiDAR branch ($\mathcal{B}_{\mathtt{L}}$), $f_{\theta^{\mathtt{L}}}^{\nu}(.)$; and (d) integration branch ($\mathcal{B}_{\mathtt{IN}}$), $f_{\theta^{\mathtt{IN}}}^{\nu}(.)$.

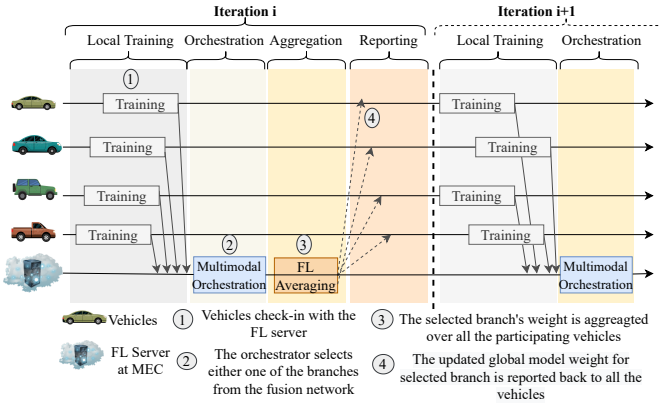


Fig. 2: In FLASH, multimodal FL training, orchestration, aggregation, and reporting occupy specific time windows in each iteration.

C. Multimodal Federated Training

The federated training architecture is composed by the local model training at the edge and federated aggregation deployed at the MEC. The global optimization of the local models requires the vehicles to periodically exchange and synchronize the model parameters $\theta_\nu^{\mathtt{FN}}$. However, these parameter exchanges and synchronizations impose overhead in both the uplink and downlink control channels, calculated as: $\widetilde{o_{ul}} = \sum_{i=1}^{\mathcal{N}} V_i \times (|\theta_\nu^{\mathtt{FN}}|)$, and $\widetilde{o_{dl}} = \mathcal{N} \times (|\theta_\nu^{\mathtt{FN}}|)$ *float32* variables, where $|\theta_\nu^{\mathtt{FN}}| = |\theta_\nu^{\mathtt{C}}| + |\theta_\nu^{\mathtt{I}}| + |\theta_\nu^{\mathtt{L}}| + |\theta_\nu^{\mathtt{IN}}|$, $\mathcal{N}$ is the total number of federated iterations, and V_i is the number of participating vehicles in the i^{th} iteration.

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 320 Mb of uplink and downlink channels. To address this problem, FLASH transmits the fusion network to the MEC in the uplink control channel with overhead of $\widetilde{o_{ul}} = |\theta_\nu^{\mathtt{FN}}|$ *float32* variables. We design a *multimodal orchestrator* at the MEC, which retrieves four branches ($\mathcal{B}_{\mathtt{C}}$, $\mathcal{B}_{\mathtt{I}}$, $\mathcal{B}_{\mathtt{L}}$, $\mathcal{B}_{\mathtt{IN}}$,) from the received network and stochastically selects one branch to be aggregated. The updated branch is then sent back through the downlink transmission. This lowers the overhead in the downlink channel to $\widetilde{o_{dl}} = \mathcal{N} \times (|\theta_\nu^{\mathcal{B}_b(i)}|)$ *float32* variables, $b \in \{\mathtt{C}, \mathtt{I}, \mathtt{L}, \mathtt{IN}\}$.

•**Algorithm for multimodal federated training:** In Alg. 1, we initialize the overall fusion network with the weights from the previous iteration at each vehicle (random initialization is used at first iteration). We define update rate for GPS coordinates, image, LiDAR, and integration branches according to a probability distribution $\mathcal{P} = \{\alpha, \beta, \gamma, \delta\}$, $\alpha + \beta + \gamma + \delta = 1$, where the parameters α, β, γ, δ denote the probability of selecting the GPS, Image, LiDAR and integration branches for aggregation, respectively. For each federated iteration i, ranging from 1 to $\mathcal{N}$, we perform local training using the model with the weights from earlier iteration, $\theta_\nu^{\mathtt{FN}(i-1)}$, for ζ epochs, and generate updated model weights $\theta_\nu^{\mathtt{FN}(i)}$. Next, the MEC assigns four different branches within the current

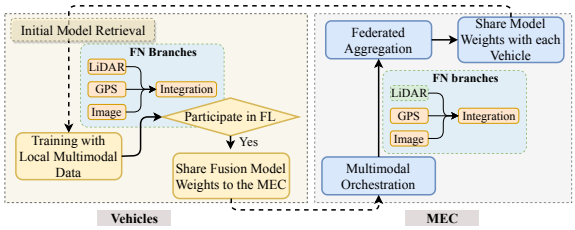


Fig. 3: The synergic local and multimodal federated training. The local training happens on the multimodal data, the orchestrator selects one of the branches from the fusion network (FN) for federated averaging (highlighted 'LiDAR' for example) in each iteration.

model weights $\theta_{\nu}^{\mathtt{FN}(i)}$ and chooses one of them $\mathcal{B}(i)$ using the stochastic function $\mathbf{P}_{\mathcal{P}}(.)$. The weights of the selected branch $\mathcal{B}(i)$ of each received model are averaged and sent back to the participating vehicles. We use straightforward averaging of the weights as our federated aggregation method [8]. The vehicles update the selected branch of their local models and execute the local training for the next federated iteration. The problem of sector selection is restricted to a fixed candidate set, making the local data independent and identically distributed (IID).

•**FL protocol in FLASH:** In general, the federated training consists of *local training*, *aggregation*, and *reporting*. However, for handling multimodal data, an *orchestration* module is added between the local training and aggregation steps of the FL protocol flow. In this orchestration step, we perform the stochastic selection of a specific branch as discussed in Alg. 1. Our overall operation over consecutive iterations is shown in Fig. 2, with the time windows for the local training, multimodal orchestration, federated aggregation, and reporting displayed. The time window for each step is defined based on the application requirements.

•**FL training in FLASH:** An overview of the proposed multimodal FL training is presented in Fig. 3. The initial model retrieval block is used to download the most updated global model from the MEC to the new vehicle as it comes within the coverage of the BS associated with the MEC. Each vehicle performs local training on the local multimodal data for a few epochs and determines whether to participate in the global optimization. If a vehicle decides to participate, the vehicle broadcasts the model weights for the overall fusion network (encapsulating four branches, GPS, image, LiDAR, and integration) to the MEC. Meanwhile, the orchestrator at the MEC selects one of the branches as a candidate for federated averaging and transmits back the aggregated weights of the selected branch to the participating vehicles.

D. Resilient Inference

In FLASH, a vehicle receives the globally updated multimodal fusion architecture from the MEC. This model requires inputs from all sensor modalities at any given time. However, this may not be possible due to hardware or software malfunctions that may impair data availability from a specific sensor at a given time. A classical neural network may fail to handle such situations with missing input. Thus, we design a multimodal data adaptation technique that compensates the missing data from a given sensor with time-shifted copies of earlier data from the same sensor. By using historical information, we enable *resilient inference*, with graceful performance degradation. We present the pipeline of the proposed data adaptation method in Fig. 4. If a sensor data type is unavailable at a particular time instance, the 'loop-back' block finds the last available historical data for that sensor and uses that for inference.

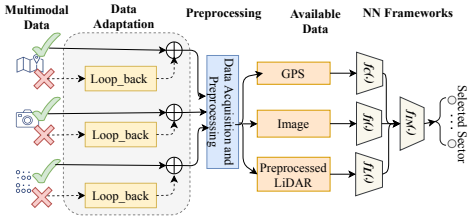


Fig. 4: The inference pipeline for sector prediction at each vehicle. The proposed inference engine enhances the trained neural network models with the added feature of adaptation to missing information.

V. FLASH Testbed Setup

We validate FLASH with experimental data (published in [10]) collected from an actual autonomous car with multimodal sensors, and mounted with programmable IEEE 802.11ad Talon Routers that operate in the 60 GHz band.

A. Testbed Environment and Sensors

We demonstrate FLASH in a scenario that resembles an urban canyon. We set up our testbed on days with dry, low humidity weather conditions in a metropolitan city on a two-way paved alleyway between two high-rise buildings, as presented in Fig. 5 (a). The exteriors of the buildings, which are made of brick, metal, and glass, are located at least 4 ft (1.2 m) from either side of the road. There are a few small trees and shrubs planted between the buildings on the sidewalk.

(a) (b)

Fig. 5: (a) Top view of location; (b) experimental setup.

1) Choice of Sensors: The sensor suite consists of a camera, LiDAR, and GPS, which are all attached to a 2017 Lincoln MKZ Hybrid autonomous car. The camera system consists of one GoPro Hero4 with a field-of-view (FOV) of 130 degrees. The LiDAR system consists of two Velodyne VLP 16 LiDARs with a FOV of 360 degrees. The car is equipped with an on-board computer connected to the LiDAR and GPS sensors, as shown in Fig. 5 (b). The data is captured at the following rates: 1 Hz for GPS, 30 frames per second (fps) for the camera, 10 Hz for LiDAR, and 1-1.5 Hz for the RF ground truth. Possible errors in GPS accuracy do not affect our system as long as the relative positions of the vehicle during trials are maintained.

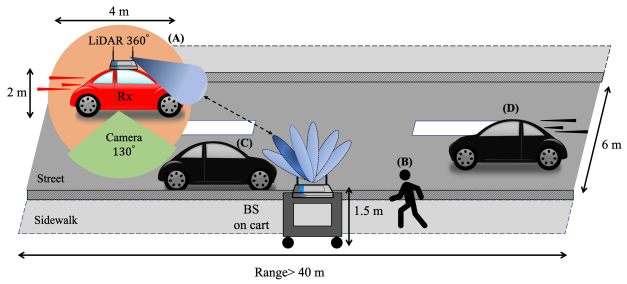


Fig. 6: Schematics of data collection environment for: A) Category 1: LOS passing, B) Category 2: NLOS pedestrian, C) Category 3: NLOS static car, D) Category 4: NLOS moving car.

Cat.	Spd. (mph)	Lane	Featuring	Scenarios	# Eps.	# Smpl.
1	10, 15, 20	same opposite	-	–	60	9729
2	15	opposite	pedestrian	standing walk right to left walk left to right walk back to front walk front to back	50	7968
3	15, 20	opposite	static car	on right on left in front	60	8174
4	15, 20	opposite	moving car	10mph same lane 20mph same lane 10mph opposite lane 20mph opposite lane	40	6052

TABLE I: Summary of different categories of collected dataset.

2) mmWave Radios: We use TP-Link Talon AD7200 tri-band routers, which use Qualcomm QCA9500 IEEE 802.11ad Wi-Fi chips with an antenna array to work as both the BS and Rx at the 60 GHz frequency [4]. The default codebook includes sector IDs from 1 to 31 and 61-63 for a total of 34 sectors; the sectors with IDs of 32 to 60 are undefined. We gain access to PHY-layer characteristics of AP and RX using the open-source Linux Embedded Development Environment (LEDE) and Nexmon firmware patching released by [4], [11]. We record the time-synchronized RF ground truth data as data transmission rate and received signal strength indication (RSSI) at each sector.

B. Testbed Settings

We define four different categories as: (a) LOS passing, (b) NLOS with a pedestrian in front of the BS, (c) NLOS with a static car in front of the BS, and (d) NLOS with a car moving between the Rx and the BS (see Fig. 6) with additional variations as shown in Tab. I. For each scenario, we collect 10 *episodes*, or trials, with episode durations of approximately 15 seconds. We limit the vehicle's speed to 20 mph, which is typical for inner-city roads.

1) Image Extraction from Videos: For each of the videos collected with the GoPro we use the OpenCV python library and split up each video into its individual frames and save each frame as an image with corresponding system timestamps. As an example, for a 15 second video with a frame rate of 30 fps, we obtain around 450 images.

2) Synchronization: We note that among the mounted sensors, the camera has the highest sampling rate at 30 fps, whereas LiDAR and GPS have 10 Hz and 1 Hz rates, respectively. According to the 802.11ad standard, sector sweeping is repeated whenever a drop in the received signal power

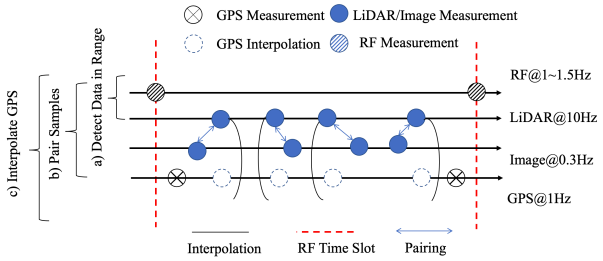


Fig. 7: Synchronization scheme.

is observed at the Rx, which is an indication of a sector misalignment. As the optimum sector does not change between two consecutive RF measurements, we up-sample our ground truth RF data measurements by associating the same optimum sector to non-RF data between two consecutive RF samples. In particular, our synchronization scheme has three steps as shown in Fig. 7: for each time slot between two RF samples; a) detect the LiDAR and image sensor data within the corresponding time slot; b) pair each LiDAR sensor data with the closest image and record the timestamp; c) for each timestamp, interpolate the GPS coordinates and record the RF ground truth data. For GPS interpolation, assuming that the car is moving at a constant speed, we first estimate the GPS coordinates at the time that RF samples are recorded for the target time slot. We then detect the GPS coordinates of the two closest points, say, $(\text{lat}_1, \text{lon}_1), (\text{lat}_2, \text{lon}_2)$, and estimate the coordinates at the RF sample timestamp $(\text{lat}_\text{x}, \text{lon}_\text{x})$ as:

$$\text{lat}_\text{x} = \begin{cases} \frac{n\ \text{lat}_1 + m\ \text{lat}_2}{n+m} & if \quad \text{lat}_1 < \text{lat}_\text{x} < \text{lat}_2 \\ \frac{n\ \text{lat}_1 - m\ \text{lat}_2}{n-m} & o.w. \end{cases} \tag{4}$$

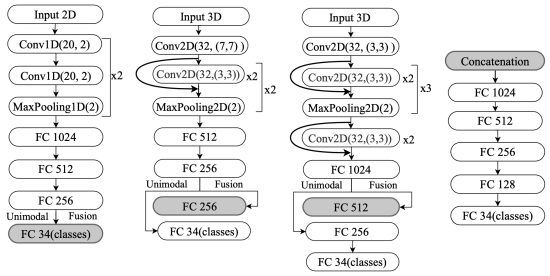
where $m = |t_{\text{lat}_1} - t_{\text{lat}_\text{x}}|$ and $n = |t_{\text{lat}_2} - t_{\text{lat}_\text{x}}|$. The same equations are used to estimate the longitude.

VI. Experimental Analysis

In this section, we validate our proposed federated architecture on the FLASH dataset. For experiments, we use Keras 2.1.6 with Tensorflow backend (version 2.2.0).

A. Experiment setting

1) Dataset: To evaluate the FLASH framework, 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 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 separating 80% and 10% of the overall *local dataset*. 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.



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

Fig. 8: Proposed network architectures for (a) GPS, (b) image, (c) LiDAR, and (d) integration networks. We use multiple convolutional and fully connected (FC) layers. The integration model is designed by concatenating the highlighted layers from each unimodal model.

2) Implementation Details: For all models (see Fig. 8), we exploit categorical cross-entropy loss for training with a batch size of 32 for 100 epochs. We use Adam [21] as our optimizer with $\beta = (0.9, 0.999)$ and initialize the learning rate to 0.0001. We set the LiDAR range to be within ± 80 m. We quantize each axis to a (20, 20, 20) block array which correspond to steps of (2.79, 4.65, 0.5). Moreover, we resize the high quality raw images to (160, 90, 3) for input.

3) Performance Metrics: Top-K accuracy is the percentage of times that the model includes the correct prediction among the top-K probabilities. The errors in prediction, i.e., selecting a sub-optimal sector, can affect the system 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.

B. Competing Methods

We compare our proposed FLASH framework against two other DL approaches in accuracy and overhead in Sec. VI-F.

• **Local Learning and Global Inference:** 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 and Global Inference:** The vehicles participate in a data sharing scheme to converge to a generalized model. As a result, V 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 back channel with the required bandwidth for sharing such large amounts of data.

• **FL and Global Inference (FLASH):** The vehicles use only their local training data to optimize their local model. Each vehicle participates in a global model aggregation round, where only the local models are sent to the MEC.

C. Local Learning and Global Inference

In the first set of our experiments, we train DNNs on the local dataset for each vehicle, separately. During inference, we use the global test dataset to compare the performance. We demonstrate the average achieved accuracy over all 10 vehicles in Fig. 9 for different categories. Results reveal that each model trained on the local dataset fails to achieve competitive performance when exposed to the global test dataset in the inference phase. Additionally, we observe that both the image and fusion networks give better prediction accuracy than GPS or LiDAR in most cases. Thus, we choose *fusion* as our selected architecture for the rest of our experiments, as it enables resilient inference as well. Even though the top-1 accuracies are in the lower range in Fig. 9, it performs comparably better than the random selection accuracy of 0.029 (1 among 34 classes). We observe the top-5 and top-10 accuracies vary in the range of 40%-60% and 50%-75%, respectively.

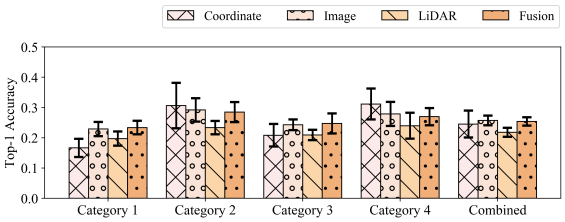


Fig. 9: Average achieved top-1 accuracy of local training and global inference over all vehicles. The error bars depict the variance in top-1 accuracies among all vehicles.

Vehicles	Top-1 Accuracy (%)					
	Cat. 1	Cat. 2	Cat. 3	Cat. 4	Combined	**Incr. Boost**
1	22.87	28.13	24.49	33.38	26.58	-
2	29.37	37.14	33.37	38.22	34.01	**7.43**
3	40.05	43.84	40.47	42.90	41.64	**7.63**
4	46.85	49.81	47.21	52.74	48.79	**7.15**
5	52.04	57.12	53.49	62.09	55.58	**6.79**
6	62.03	64.79	62.36	65.80	63.52	**7.94**
7	68.13	72.10	68.75	70.64	69.75	**6.23**
8	73.62	78.07	74.43	79.67	76.08	**6.33**
9	80.91	83.55	80.47	84.19	82.08	**6.00**
10	85.91	90.37	85.32	88.22	87.31	**5.23**

TABLE II: The top-1 accuracy while training on local dataset of $\mathcal{V} = 1, ..., 10$ vehicles and testing on global test set.

D. Centralized Learning and Global Inference

In this set of experiments, we explore the effects of centralized learning on global test data. Considering the local training dataset available at each vehicle, we construct an *accumulated training set* by gathering the local training set from V vehicles. We then train the model using the accumulated training set and test it on the global test set. We present the result of this experiment in Tab. II, where 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. We observe a surge in top-1 accuracies as we keep adding more vehicles to the accumulated training set. The incremental improvement after adding one more vehicle is highlighted in the last column. Although this approach improves the robustness of sector selection, it requires all the training data to be gathered at one site (unlike FLASH), e.g., a cloud, with associated transmission cost and privacy concerns.

E. Federated Learning and Global Inference (FLASH)

In our proposed Alg. 1, the vehicles participate in federated aggregation, where different branches of the models are

selected through a multimodal orchestrator. The aggregation policy is based on a stochastic function $\mathbf{P}_{\mathcal{P}}(.)$, captured by parameters: $\alpha, \beta, \gamma, \delta$ for GPS, image, LiDAR, and integration networks, respectively. We provide a comprehensive study on the effect of different policies on top-1 global accuracy. First, with LiDAR being the most successful unimodal network, we define a *greedy LiDAR* policy where only the LiDAR branch is aggregated, denoted as $\mathcal{P}_{\text{Greedy LiDAR}}$. In the second policy $\mathcal{P}_{\text{L,IN Biased}}$, we bias the LiDAR and integration branches and select the LiDAR and integration branches with probability of 0.4 and the GPS and image branches with probability of 0.1. Next, we consider an *unbiased* policy $\mathcal{P}_{\text{Unbiased}}$ where one branch is selected randomly following a uniform distribution. These policies are parameterized as follows:

$$\mathcal{P}_{\text{Greedy LiDAR}} = (0, 0, 1, 0)$$
$$\mathcal{P}_{\text{L,IN Biased}} = (0.1, 0.1, 0.4, 0.4)$$
$$\mathcal{P}_{\text{Unbiased}} = (0.25, 0.25, 0.25, 0.25)$$

We also consider a final policy $\mathcal{P}_{\text{All}}$ where the entire model of the fusion network is averaged and updated without orchestrating any specific branch. We assume all 10 vehicles participate in FL aggregating and run 100 iterations.

Fig. 10a denotes the improvement in global top-1 accuracy achieved by multiple rounds of aggregating the model weights following the above policies. Although model aggregation improves results for all policies, we observe the lines converge after 50 iterations. In particular, the maximum top-1 accuracy following the policy $\mathcal{P}_{\text{All}}$ is 68.17%. On the other hand, $\mathcal{P}_{\text{Greedy LiDAR}}$, $\mathcal{P}_{\text{L,IN Biased}}$, $\mathcal{P}_{\text{Unbiased}}$ policies achieve the top-1 accuracy of 39.42%, 52.23%, and 59.72%, respectively. The size of GPS, image, LiDAR, and integration model branches are 2.78MB, 26.55MB, 3.73MB, and 6.21MB, respectively. As a result, the corresponding overheads for $\mathcal{P}_{\text{Greedy LiDAR}}$, $\mathcal{P}_{\text{L,IN Biased}}$, $\mathcal{P}_{\text{Unbiased}}$, and $\mathcal{P}_{\text{All}}$ policies are 3.73MB, 6.90MB, 9.81MB, and 39.27MB on average, respectively. In other words, even though the $\mathcal{P}_{\text{All}}$ policy yields to best top-1 accuracy, it imposes 9.52x, 4.69x, and 3x extra overhead than the $\mathcal{P}_{\text{Greedy LiDAR}}$, $\mathcal{P}_{\text{L,IN Biased}}$, $\mathcal{P}_{\text{Unbiased}}$ policies, respectively. Hence, the branch selection policy gives the flexibility to use less wireless resources to adhere to user-imposed constraints, such as a threshold on the allowable data-rate over downlink, which is easier to maintain when sending only one branch instead of the entire fusion network.

F. Accuracy and Overhead Trade-off

In this section, we first compare the accuracies of three competing methods, presented in Fig. 10b. The performance of the local learning method, Sec. VI-C, is denoted with a diamond marker. The dashed line indicates the improvement achieved by centralized learning between multiple vehicles at the cost of transmitting all the data to a central unit. The star, dot, and triangle markers show the FL results at iterations 10, 40, and 78. We observe that in order to achieve 68.17% top-1 accuracy, the centralized learning requires data from around 7 vehicles, while the FL can achieve the same accuracy without data sharing and with only 78 rounds of aggregation.

Methodology	Acc.(%)	Overhead(s)	
		Data sharing	Model sharing
Local Learning	36.78	-	-
Centralized Learning	87.31	11.54	0.1813
FLASH (78 iterations)	68.17	-	3.51

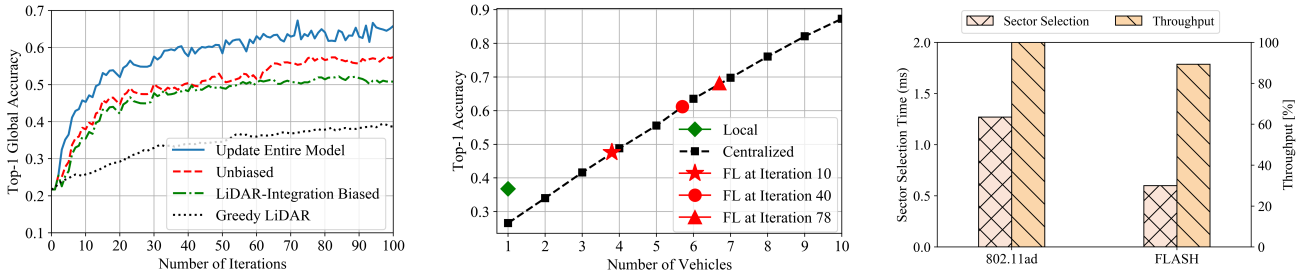
TABLE III: Comparing the performance of the three data-driven competing methods with respect to accuracy and model initialization overhead. All accuracies are reported on the global test set.

However, both the centralized and federated methods impose some communication overhead in the control channel for *model initialization*. We observe a trade-off between overhead and accuracy for both the methods, presented in Tab. III. Though the local learning approach does not require any data/model sharing, it provides up to only 36.78% top-1 accuracy. On the other hand, the centralized learning approach can provide 87.31% accuracy, but it comes with a large communication cost of transmitting the entire data (2.5 GB) to the cloud, as well as privacy concerns. Meanwhile, FLASH reduces the communication cost while preserving 68.17% accuracy without any sort of data sharing. FL aggregation iterations continue in the background and do not disrupt the inference. In each aggregation round, one out of four branches is sent back to the vehicles with 696,600 parameters for the lightest branch, i.e., for GPS, and 6,638,368 parameters for the heaviest one, i.e., for images. The back channel is supported by the 5GHz band of the Talon router with a data rate of 1733Mbps. Thus, it takes 45ms on average to retrieve the model in each iteration with unbiased policy, considering 32 bits per model parameter (314.31Mb overall). For a total of 78 aggregation rounds, the model initialization overhead sums up to 3.51 seconds. Hence, we conclude that FLASH provides a 46.04% improvement in accuracy over local learning and a 70.05% improvement in overhead over centralized learning.

G. Sector Selection Speed and Throughput Ratio

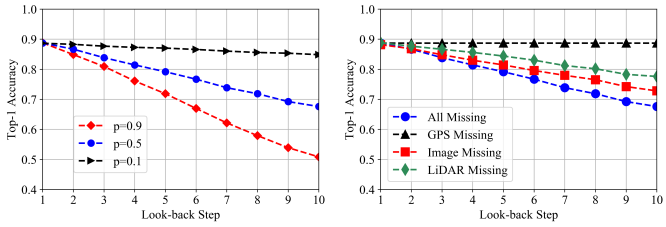
Once we establish that FLASH outperforms the other two competing methods in terms of both accuracy and overhead, we next compare the sector selection speed against the current mmWave standards. As described in Sec. III-A, in the traditional exhaustive search approach, each sector of BS transmits M probes to initiate the communication. The end-user then returns the optimal sector ID to the BS.

FLASH infers 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; (b) *Preprocessing*: the LiDAR preprocessing step described in section IV-A has a negligible latency that can be further reduced by exploiting parallel processing; (c) *Model inference*: we pass a test sample 100 times over the DL model and calculate the average inference delay of 0.6 ms; (d) *Sector sharing*: an integer varying between 0-31 and 61-63, representing the sector ID with 7 bits, is sent back to the paired users. Considering the 5GHz back channel of the Talon routers, transmitting the optimal sector back takes only 4 ns. As a result, the FLASH inference consumes $\sim$0.6 ms end-to-end. On the other hand, sweeping all 34 sectors with the 802.11ad standard in Talon



(a) (b) (c)

Fig. 10: (a) The performance of federated training and global inference over 90 rounds of aggregation. (b) Comparing the performance of FL with an increasing number of vehicles and amount of federated training. (c) Comparing the performance of 802.11ad and FLASH with respect to throughput ratio and end-to-end sector selection time.



(a) (b)

Fig. 11: (a) Performance of resilient inferencing when all three modalities are missing with probability $p = 0.1, 0.5, 0.9$. We loop back the sensor data to different sample values presented in the x-axis. (b) Tolerance of the proposed FLASH framework when different combinations of sensors modalities are missing.

routers takes 1.27 ms [4]. We also calculate the throughput ratio (defined in Sec. VI-A3) of FLASH and the 802.11ad standard in Fig. 10c. We observe 52.75% improvement in sector selection speed while retaining 89.32% throughput ratio.

H. Resilient Inference for Missing Sensors

To evaluate the resiliency of FLASH, we first consider the extreme scenario in which all the sensor data are missing. We evaluate the performance with respect to the parameters defined in Sec. IV-D, namely the loop-back step and probability of missing data in Fig. 11a. We observe that the top-1 accuracy is resilient to the low loop-back steps and decreases as the loop-back step increases and the samples become far apart. However, the higher probability of missing information results in lower top-1 accuracy. Next, we explore the effect of missing different combinations of sensor data in Fig. 11b, with a fixed probability of 0.5. We observe the absence of GPS negligibly affects the performance. One might argue that the LiDAR preprocessing step described in Sec. IV-A requires the location of the vehicle as input; however, the coordinates can also be estimated using the GPS interpolation scheme presented in Eq. 4. Note that two scenes separated by a small amount of time might result in the same LiDAR data due to quantization, while images are completely different. From Fig. 11b, we observe that FLASH can retrieve 67.59% top-1 accuracy when up to ten samples are missing for all modalities.

I. Comparison with State-of-the-art

In Tab. IV, we benchmark the performance of our proposed FL architecture against the state-of-the-art DL-based approaches by Klautau *et al.* [16] and Dias *et al.* [17]. 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 [22]. 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 [14], [23], [24], [25], [26]; (b) consideration of LOS-only scenarios while using camera sensors [27], [28]; and (c) inclusion the RF inputs (sub-6 GHz channel measurements, for instance) [15]. In Tab. IV, we observe that FLASH outperforms the state-of-the-art by 35-45% in top-1 accuracy.

Methods	Modalities	Architecture	Top-1 Acc. (%)	Dataset	Evaluation Type	Task
Klautau *et al.* [16]	LiDAR	Centralized	30.5 ± 1	Synthetic Raymob-time [22]	Simulation	Beam Prediction
Dias *et al.* [17]	LiDAR	Centralized	20.5 ± 1	Synthetic Raymob-time [22]	Simulation	Beam Prediction
FLASH	GPS, Image, LiDAR	Distributed	**68.17**	FLASH Dataset	Testbed	Sector Selection

TABLE IV: Comparison of FLASH with state-of-the-art techniques which use non-RF data for similar tasks.

VII. Conclusions

We make a case for using multiple sensor modalities [29] to aid in mmWave beamforming, as opposed to using only RF-based approaches. FLASH incorporates multimodal data fusion using DL architectures, whose training and dissemination in real-world vehicular networks, as well as resilience to missing data sources, can be practically achieved using a FL architecture. Results obtained on datasets collected by an autonomous vehicle with LiDAR, GPS, and camera sensors indicate 52.75% reduction for mmWave sector selection time while retaining 89.32% of throughput as compared to the traditional sector sweeping. The dataset and the code for the proposed fusion models in FLASH are released online in [10] for independent validation and further research on distributed multimodal learning.

Acknowledgement

The authors gratefully acknowledge the support from Chris Dick (NVIDIA Corporation), funding from the US National Science Foundation (grants CCF-1937500 and CNS-2112471), and Northeastern University's Field Robotics lab.

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] 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.

[5] 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.

[6] 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*, 2007, pp. 1044–1049.

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

[8] 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.

[9] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork, "Weather Influence and Classification with Automotive Lidar Sensors," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1527–1534.

[10] "FLASH Dataset," https://genesys-lab.org/multimodal-fusion-nextg-v2x-communications.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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)*, 2019, pp. 1–5.

[18] 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 *IEEE International Conference on Mobility, Sensing and Networking*, Dec 2021.

[19] 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.

[20] Tutorialink: Convert Latitude and Longitude to X and Y Grid System" https://python.tutorialink.com/convert-latitude-and-longitude-to-x-and-y-grid-system-using-python/.

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

[22] 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)*, 2018, pp. 1–9.

[23] Y. Wang, A. Klautau, M. Ribero, M. Narasimha, and R. W. Heath, "MmWave Vehicular Beam Training with Situational Awareness by Machine Learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[24] 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)*, 2016, pp. 1–6.

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

[26] B. Salehi, M. Belgiovine, S. G. Sanchez, J. Dy, S. Ioannidis, and K. Chowdhury, "Machine Learning on Camera Images for Fast mmWave Beamforming," in *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, 2020, pp. 338–346.

[27] Y. Tian, G. Pan, and M.-S. Alouini, "Applying Deep-Learning-Based Computer Vision to Wireless Communications: Methodologies, Opportunities, and Challenges," *IEEE Open Journal of the Communications Society*, 2020.

[28] 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.

[29] D. Roy, Y. Li, T. Jian, P. Tian, K. R. Chowdhury, and S. Ioannidis, "Multi-modality Sensing and Data Fusion for Multi-vehicle Detection," *IEEE Transactions on Multimedia*, 2022.