This is a pre print version of the following article:

Federated mmWave Beam Selection Utilizing LIDAR Data / Mashhadi, M.B., Jankowski, M., Tung, T.-Y., Kobus, S., Gunduz, D.. - In: IEEE WIRELESS COMMUNICATIONS LETTERS. - ISSN 2162-2337. - 10:10(2021), pp. 2269-2273. [10.1109/LWC.2021.3099136]

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

17/11/2026 17:51

(Article begins on next page)

Federated mmWave Beam Selection Utilizing LIDAR Data

Mahdi Boloursaz Mashhadi*, Mikolaj Jankowski*, Tze-Yang Tung, Szymon Kobus, Deniz Gündüz
Dept. of Electrical and Electronic Eng., Imperial College London, UK

arXiv:2102.02802v2 [cs.IT] 25 Jul 2021

***Abstract*—Efficient link configuration in millimeter wave (mmWave) communication systems is a crucial yet challenging task due to the overhead imposed by beam selection. For vehicle-to-infrastructure (V2I) networks, side information from LIDAR sensors mounted on the vehicles has been leveraged to reduce the beam search overhead. In this letter, we propose a federated LIDAR aided beam selection method for V2I mmWave communication systems. In the proposed scheme, connected vehicles collaborate to train a shared neural network (NN) on their locally available LIDAR data during normal operation of the system. We also propose a reduced-complexity convolutional NN (CNN) classifier architecture and LIDAR preprocessing, which significantly outperforms previous works in terms of both the performance and the complexity.**

***Index terms*— Federated learning, mmWave beam selection, LIDAR.**

I. Introduction

Millimeter wave (mmWave) is a promising technology for high data rate vehicle-to-infrastructure (V2I) communications. However, efficient beam selection in mmWave communications is challenging due to the overhead imposed by the beam search process. Recently it was shown that side information from sensors mounted on vehicles can be exploited to reduce the beam-selection overhead for mmWave links. For instance, position information can be used to query the most prominent mmWave beams [1]. Inertial sensors placed on vehicle's antenna arrays enable efficient antenna element configuration by tracking the orientation of the vehicle [2]. Furthermore, position and motion information can be jointly processed to further reduce the alignment overhead [3]. From the infrastructure side, a radar located at the base station (BS) can help estimate the direction of arrival and aid the beam search [4]. Spatial information obtained from out of band measurements is exploited in [5]–[8] where [7], [8] used sub-6GHz channel measurements to train neural network (NN)s for mmWave beamforming. Vision-aided approaches are considered in [9]–[11]. BSs equipped with cameras are proposed to employ computer vision and deep learning techniques to predict mmWave blockage and beam strength in [10]. The authors in [11] build a panoramic point cloud from images taken within the cellular coverage area. This point cloud is then input to a neural network (NN) to predict the optimal beams.

The use of light detection and ranging (LIDAR) technology is considered in [12], [13], where a NN architecture is trained over simultaneous LIDAR and ray-tracing channel datasets to identify K beam directions that include the beam pair with the best channel condition between the vehicle and the BS with the highest probability. The approach in [12], [13] is distributed, in the sense that, each vehicle uses the trained NN on the measurements from its own LIDAR sensor to infer its top-K beam directions. It is shown in [13] that such a distributed approach outperforms centralized beam selection, where a NN at the BS infers the best beams for all the vehicles in its coverage area either by combining LIDAR data from all the vehicles or using a single LIDAR sensor mounted at the BS. Although the NN performs beam selection inference in a distributed fashion in [12], [13], it is trained offline on LIDAR and channel measurements from all the vehicles gathered in a centralized dataset. However, in practice, gathering a large centralized dataset of individual LIDAR measurements from vehicles in each individual cell is challenging as it requires communicating a large amount of LIDAR point cloud data over the uplink channel.

This paper builds upon the unpublished work of the authors that recently won the "AI/ML in 5G" competition ranking second in "ML for mmWave beam selection" challenge organized by the International Telecommunications Union (ITU) [14], [15].In this work, we propose fully a distributed LIDAR-aided beam selection method for V2I mmWave communication systems, in which both the inference and training of the NN are performed in a distributed fashion at the vehicles in the coverage area of the BS. We propose a three-phase procedure, which enables the vehicles to periodically collect up-to-date data and train or fine-tune the NN in a federated manner during normal operation of the system. After the training phase, each vehicle leverages the trained NN and its locally available LIDAR data to infer a subset of beams that are most likely to contain the best transmitter/receiver beam pair. We also propose a reduced-complexity convolutional NN (CNN) architecture along with LIDAR preprocessing, which significantly outperforms previous works. The proposed architecture achieves a top-10 classification accuracy of 91.17% on the benchmark Raymobtime dataset [16], which is a significant improvement over the previous works in [12], [13], while reducing the number of floating point operations (FLOPs) and parameter complexity of the NN by factors of 100 and 55, respectively. For further reproduction of the reported results, our codes are available at: https://github.com/galidor/ITU_Beam_Selection_TF

II. System Model

We consider a downlink orthogonal frequency division multiplexing (OFDM) mmWave system, where a BS located on the street curb serves connected vehicles in its coverage area over N_c subcarriers. The BS and the vehicles are equipped with N_t and N_r antennas, respectively. Denote by $\mathbf{H}_n$ the downlink channel matrix from the BS to a vehicle over the n'th subcarrier. We assume that both the BS and the vehicle have antenna arrays with only one radio frequency (RF) chain and apply analog beamforming. We assume fixede beam codebooks $\mathcal{C}_t = \{\mathbf{f}_i\}_{i=1}^{C_t}$ and $\mathcal{C}_r = \{\mathbf{w}_j\}_{j=1}^{C_r}$ at the transmitter and receiver sides, respectively.

Utilizing a pair $(i,j) \in \mathcal{C}_t \times \mathcal{C}_r$ of precoder and combiner vectors, the resulting channel gain at subcarrier n is $\mathbf{w}_j^H \mathbf{H}_n \mathbf{f}_i$, where $(\cdot)^H$ denotes the conjugate transpose. For fair comparison with previous works [12], [13], we also assume a noise-free mmWave setting.For the (i,j) pair, the sum power gain over all subcarriers is given by $y_{ij} = \sum_{n=1}^{N_c} |\mathbf{w}_j^H \mathbf{H}_n \mathbf{f}_i|^2$. Hence, the optimum beam label is $b^* = (i^*, j^*) = \underset{(i,j)}{\text{argmax}}\ y_{ij}$. Without any side information, the transmitter and receiver would search through all $C_t C_r$ beam pairs to identify b^*. Our goal is to infer a small subset of K beam pairs $\mathcal{S} = \{(i_k, j_k)\}_{k=1}^{K} \subset \mathcal{C}_t \times \mathcal{C}_r$ using the available position and LIDAR data, such that $b^* \in \mathcal{S}$. This results in a reduction of $\frac{K}{C_t \times C_r}$ in the search space for beam selection. In the next section, we propose a novel NN architecture as well as a federated training approach for top-K beam classification from simultaneous position and LIDAR data.

* Indicates equal contribution.

This work was supported by the European Research Council (ERC) through project BEACON (grant no 677854).

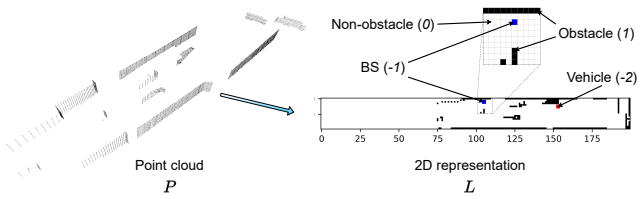


Fig. 1: Preprocessing of the LIDAR point cloud.

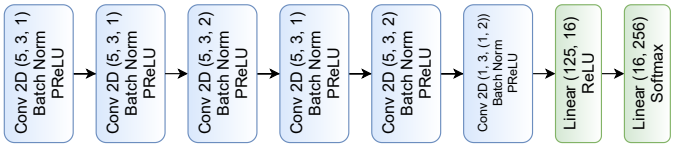


Fig. 2: The proposed model architecture.

III. Federated Beam Selection Utilizing LIDAR Data

We propose a novel data-driven beam selection scheme, where connected vehicles in the coverage area of a BS collaborate to train a shared NN for top-K beam classification using their position and LIDAR data in a distributed manner. Collaborative training is orchestrated by the BS, and takes place during normal operation of the network.

A. Three-Phase Network Operation

Our proposed solution consists of three network operation phases: (i) data collection phase, (ii) federated training phase, and (iii) distributed inference phase.

During phase (i), a subset of connected vehicles in the coverage area of the BS, denoted by $\mathcal{V} = \{v\}_{v=1}^{V}$, each acquires a local dataset $\mathcal{D}_v = \{(\mathcal{P}_v, \mathcal{B}_v)\}_{v\in\mathcal{V}}$, where $\mathcal{P}_v = \{P_i\}_{i=1}^{|\mathcal{D}_v|}$ contains instances of the point cloud P_i recorded by the LIDAR sensor and $\mathcal{B}_v = \{b_i^*\}_{i=1}^{|\mathcal{D}_v|}$ contains the corresponding best beam pair labels $b_i^* \in \mathcal{C}_t \times \mathcal{C}_r$, i.e., index of the best beam pair. During this phase the vehicles employ any beam selection or tracking technique of their choice to identify the best beam pair.

During phase (ii), vehicles with local datasets collaborate in a federated learning scheme to train or fine-tune a shared NN for top-K beam classification. In particular, the vehicles employ federated averaging (FedAvg) [17], where a global model is sent to the vehicles by the BS at each round, and the vehicles perform mini-batch stochastic gradient descent (SGD) updates based on their local datasets. The local updates are aggregated by the BS, and used to update the global model for the next round. The duration of this phase is proportional to the number of global aggregation rounds required to train the model, denoted by N_a. Note that, the vehicles train a single site-specific NN, which learns the statistical characteristics of the coverage area of the BS for efficient beam selection.

Finally, in phase (iii), any vehicle in the coverage area of the BS utilizes the most up-to-date trained/tuned NN on its local LIDAR data to infer K beams and reduce the beam search overhead. Note that, in this phase, the BS can use a low frequency control channel to transmit the trained NN model to any new vehicle entering its coverage area.

Each BS in a large network can orchestrate training of a site-specific NN for its own coverage area following the above three phases. Note that the above operation may either be initialized randomly, or from a NN pre-trained on a centralized dataset offline. As the above three phases take place periodically, the NN can automatically adapt to changes in the statistics of data over time through periodic fine-tuning of the NN with up-to-date data. Also

Algorithm 1: FedAvg for LIDAR-assisted beam selection

Init: Initial parameters $\boldsymbol{\theta}_v^{(0)} = \boldsymbol{\theta}^{(0)}$, $\forall v \in \mathcal{V}$.
for *each* $m = 1, 2, \ldots$ **do**
  Each vehicle performs a local epoch using mini-batch gradient decent iterations according to (2) ;
  if m *is an integer multiple of* N_v **then**
    Each vehicle v sends $\boldsymbol{g}_v^{(m)} = \boldsymbol{\theta}_v^{(m)} - \boldsymbol{\theta}_v^{(m-N_v)}$ to BS;
    BS computes $\boldsymbol{\theta}^{(m)} = \boldsymbol{\theta}^{(m-N_v)} + \frac{\mu}{|\mathcal{V}|}\sum \boldsymbol{g}_v^{(m)}$;
    BS distributes $\boldsymbol{\theta}^{(m)}$ such that $\boldsymbol{\theta}_v^{(m)} = \boldsymbol{\theta}^{(m)}$, $\forall v \in \mathcal{V}$;
  end
end
Output: Trained $\boldsymbol{\theta}^{(m)}$ shared among all vehicles.

note that the above phases take place during normal operation of the vehicular network and impose no interruption. The vehicles keep communicating with the BS as they would do without the above phases (utilizing any beam selection or tracking technique of their choice) but just record and collect the beam labels and the corresponding LIDAR inputs for training. Only some vehicles with sufficient dataset would transmit model updates when fine-tuning the NN is required. Once the NN is trained up to a desired accuracy, it is used for beam selection thereby significantly reducing the beam search space.

B. LIDAR and Location Preprocessing

For each scene, the LIDAR sensor mounted on each vehicle outputs a point cloud $P = \{(x_p, y_p, z_p)\}_{p=1}^{|P|}$, representing obstacles measured by the LIDAR sensor. Each vehicle v also has its own location information (x_v, y_v, z_v), and the BS location (x_{BS}, y_{BS}, z_{BS}), which is broadcast to all the vehicles. We preprocess this data to obtain a tensor of fixed size, which contains both the location and LIDAR data and is input to the NN for each scene. To reduce both the NN dimension and the computation load, we propose a two-dimensional (2D) representation of the LIDAR measurements, where we partition the coverage area of the BS into a grid of equal-size square cells from the top view (see Fig. 1). We define the corresponding 2D tensor L, where the cells containing the BS and the vehicle are set to -1 and -2, respectively, while each of the remaining cells is populated with a 1 if it accommodates at least one of the cloud points, and with a 0 otherwise. We remark that this 2D representation discards the height data along the z-axis, resulting in a significant reduction in the input size, and hence, the complexity of the NN, which in turn reduces the communication overhead for federated training.

C. NN Architecture

Our NN architecture consists of 6 convolutional layers followed by 2 linear layers as depicted in Fig. 2. In the convolutional layers, we vary the value of stride between 1 and 2, depending on whether we intend to downscale the intermediate features, or not. We apply batch normalization and parametric rectified linear unit (PReLU) activation after each convolutional layer. The first linear layer is followed by rectified linear unit (ReLU) activation, and softmax is used at the output to obtain the predictions. To achieve better generalization, convolutional layers downscale the features and ensure that only essential information is preserved. This helps avoid overfitting to the training data [18]. Note that, to reduce the communication overhead for federated training, we have minimized the trainable model parameters utilizing a convolutional structure with limited kernel sizes. We denote the NN model function by $\pi(L; \boldsymbol{\theta})$, which is a vector of length $C_t C_r$ at the softmax output. L denotes the preprocessed LIDAR and location input, while $\boldsymbol{\theta}$ denotes the trainable

TABLE I: Comparison between the proposed NN architecture and the baseline in [12], [13], both trained in a centralized manner.

Model	Top-10 accuracy	Top-10 throughput ratio	FLOPs	# of NN parameters, $\lvert\boldsymbol{\theta}\rvert$
Proposed centralized	$91.17 \pm 0.28\%$	$94.78 \pm 0.61\%$	1.72×10^6	7462
Baseline [12], [13]	$83.92 \pm 0.93\%$	$86.15 \pm 0.82\%$	179.01×10^6	403677

NN parameters. The best beam is predicted as $\hat{b^*} = \underset{b \in \mathcal{S}}{\arg\max}\, y_b$, where the prediction set $\mathcal{S}$ is given by the top-K softmax outputs.

D. Federated Training

Gathering a large dataset of LIDAR measurements from various vehicles for centralized training at the BS imposes significant communication overhead, particularly due to the large size of LIDAR point cloud measurements. Instead, we propose a federated learning approach, where the vehicles collaborate to train a single NN architecture using their local datasets [17]. To train our NN we use the empirical cross entropy loss, hence the local loss calculated at vehicle v is given by

$$\psi_v(\boldsymbol{\theta}, \mathcal{D}_v) = -\frac{1}{|\mathcal{D}_v|} \sum_{i=1}^{|\mathcal{D}_v|} \log[\pi\left(L_i; \boldsymbol{\theta}\right)]_{b_i^*}, \tag{1}$$

where $[\pi]_b$ denotes the b'th element of the model's softmax output. Each connected vehicle performs mini-batch SGD iterations to update its local vector of model parameters, denoted by $\boldsymbol{\theta}_v$, via

$$\boldsymbol{\theta}_v^{(l)} = \boldsymbol{\theta}_v^{(l-1)} - \rho_l \nabla \psi_v(\boldsymbol{\theta}_v^{(l-1)}, \{(b_{i_l}, L_{i_l})\}_{i_l \in 1,\ldots,|\mathcal{D}_v|}), \tag{2}$$

where l is the local iteration index, $\rho_l > 0$ is the local step-size, and the set $\{(b_{i_l}, L_{i_l})\}_{i_l \in 1,\ldots,|\mathcal{D}_v|}$ is a mini-batch of the local dataset with $i_l \in 1, \ldots, |\mathcal{D}_v|$. The training consists of N_v local epochs at each vehicle (i.e. N_v cycles of training on the vehicle's local dataset) and N_a aggregation rounds at the BS as summarized in Algorithm 1.

Such distributed learning orchestrated by the BS during phase (ii) requires the vehicles to periodically exchange and synchronize their local model parameters $\boldsymbol{\theta}_v$ through reliable low-rate communications with the BS. This imposes an overhead of communicating $O_{UL} = V \times N_a \times |\boldsymbol{\theta}|$ float32 variables in the uplink and $O_{DL} = N_a \times |\boldsymbol{\theta}|$ in the downlink channel. Minimizing the number of trainable parameters $|\boldsymbol{\theta}|$ is hence critical to reduce the communication overhead during phase (ii) of the network operation.

IV. Numerical Evaluations

We provide numerical evaluations on the benchmark Raymobtime datasets [16], where we train the models on samples from dataset s008 and test on those from s009. Our training dataset includes 6482 line-of-sight (LOS) and 4712 non-line-of-sight (NLOS) samples, while our test dataset includes 1473 LOS and 8165 NLOS samples, respectively (refer to [16], [19] for details on these datasets e.g. locations, frequencies, etc.).For performance comparison, we use the top-K classification accuracy defined as the probability of correctly identifying the optimal beam pair within the top-K output of the network, and the top-K throughput ratio, R, defined as $R \triangleq (\sum_{t=1}^{T} \log_2(1 + y_{\tilde{i}\tilde{j}}))/(\sum_{t=1}^{T} \log_2(1 + y_{i^* j^*}))$, where T is the number of test samples, and (i^*, j^*) and $(\tilde{i}, \tilde{i})$ denote the optimum beam pair index and the best beam pair within the top-K prediction set $\mathcal{S}$, respectively.

In Table I, we compare the performance of the NN architecture presented in Subsection III-C with the baseline architecture proposed in [12], [13], both trained in a centralized manner. In this experiment, we trained our model using the Adam optimizer [20] with an initial learning rate of 10^{-3} and batch size of 16, and train the models for 20 epochs. The grid dimensions to generate features in Section III.B is set to 20×200. In Table I, we present 95% confidence intervals for Top-10 accuracy and throughput ratio of the models calculated from 10 Monte Carlo simulations.

According to Table I, our proposed architecture not only outperforms those in [12] and [13] in terms of both the top-10 accuracy and the throughput ratio, but also significantly reduces the complexity of the model. Our architecture reduces the FLOPs and the number of trainable parameters roughly by factors of 100 and 55, respectively. Such a significant reduction in the number of trainable model parameters is specifically desirable in federated training as it leads to a significant reduction of the communication overhead.

Remember that the beam search complexity of these schemes depends on K, the size of the prediction set. Figure 3 plots the top-K accuracy and throughput ratio for the proposed and baseline architectures as a function of K, when trained in a centralized fashion. It is observed that, our proposed model architecture significantly outperforms [12], [13], e.g., to achieve a throughput ratio $R \geq 90\%$, our proposed model architecture requires $K \geq 3$ while the baseline needs $K \geq 16$. This is more than 5 times reduction in the required search space for beam selection. Also, the proposed architecture can achieve close to 80% of the optimal throughput with $K = 1$; that is, with no beam search at all.

Note that the results reported in Table I and Fig. 4 are the average values achieved on both LOS and NLOS samples. However, the NLOS case is more challenging due to blockages of the rays by other vehicles/objects. When measuring the Top-10 accuracy of our proposed approach separately on LOS and NLOS samples, we get 94.50% and 90.77%, respectively, which shows that the proposed approach performs very well on NLOS samples as well and significantly outperforms previous works [12], [13].

We also compare the performance of the proposed approach with a NN-based approach that only assumes access to the location data. Our NN architecture for the location-only approach is a 4-layer fully connected NN with 16, 32 and 16 nodes in the hidden layers, each followed by ReLU activation and batch normalization. This architecture has been hand-crafted for the best performance, and takes as input the relative location of the vehicle with respect to the BS. It achieves top-10 accuracy and throughput ratio of 87.48% and 91.96%, respectively. Hence, the performance is significantly improved when the LIDAR data is used, which is expected, as the LIDAR data gives a better understanding of the scene and includes information on objects or obstacles that cause blockage of the mmWave rays.

We next evaluate the performance of our proposed federated beam selection scheme. To generate the local dataset at each connected vehicle v, we choose $|\mathcal{D}_v| = 11000/V$ samples from the training set s008 uniformly at random, where 11000 is the total number of samples in s008. We use mini-batch SGD with an initial learning rate of 0.2 and exponential rate decay of 0.001 with a batch size of 16 for local optimization at the vehicles. We set the learning rate $\mu = 0.2$ for aggregation at the BS. We provide the performance of our proposed federated beam selection scheme in Table II. Here, we start training from a randomly initialized global model. The notation $(N_a)^{0.88}$ in this table represents the number of global aggregation rounds required for the training to achieve a top-10 accuracy larger than 88%. This is an important measure as it determines the communication overhead required to train the model to the specified accuracy. Notations $(O_{DL})^{0.88}$ and $(O_{UL})^{0.88}$ used in this table represent this

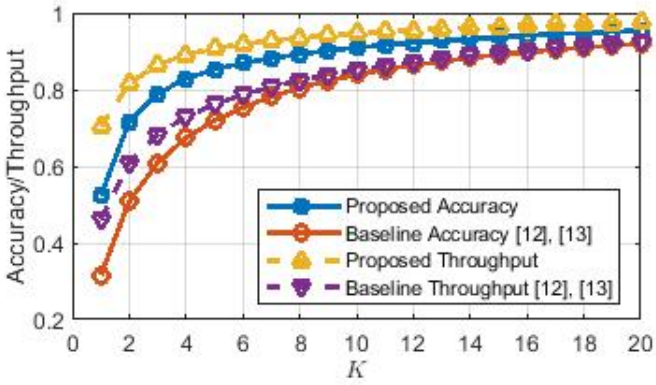


Fig. 3: Top-K accuracy and throughput ratio as a function of K.

overhead in terms of the number of float32 variables needed to be communicated over the downlink and uplink channels, respectively. According to Table II, the number of aggregation rounds required to achieve top-10 accuracy larger than 88% increases when more vehicles take part in federated training. A larger $(N_a)^{0.88}$ increases the communication overhead. However, thanks to our simple NN architecture, which only has $|\boldsymbol{\theta}| = 7462$ trainable parameters, the maximum communication overhead required for federated training (i.e., $1620 \times 7462 \sim 1.2 \times 10^7$ float32 communications for $V = 20$, $N_v = 1$) is orders of magnitude smaller than the overhead that would be imposed by offloading the LIDAR point clouds to the BS for centralized training (i.e., $\sim 4 \times 10^9$ float32 communications for samples in s008). Note that the communication overhead for federated training can further be significantly reduced utilizing gradient quantization [21]–[23], gradient sparsification [24], [25], and over-the-air aggregation [26], [27] techniques.Also note that utilizing various neural architecture search (NAS) approaches [28]–[30], we can design further simplified but well-performing NN architectures, thereby reducing the communication overhead for federated training.

Note that the size of the local dataset at a vehicle, i.e. $|\mathcal{D}_v|$, depends on how long the vehicle stays in the coverage area of the BS and how frequently it can collect samples. $|\mathcal{D}_v|$ is a design parameter to be set for each specific site depending on dimensions of the coverage area and the traffic flow speed. For the Raymobtime s008/s009 datasets used in our simulations, the coverage area is a $337 \times 202\text{m}^2$ region covering the intersection of Kent and 19th street in Rosslyn, Virginia, the average vehicle speed is $8.2\frac{\text{m}}{\text{s}}$ and the sampling period is 0.1s [19]. Hence, an average vehicle traveling along the Kent street can collect approximately $\frac{337}{8.2\times 0.1} \approx 411$ samples during its stay in the coverage area of the BS, which is approximately the local dataset size used in Table II for $V = 20$. Note that vehicles may collect more samples if they are parked, move slower, return to the cell multiple times, or have faster sampling equipment.

The last column in Table II reports the final top-10 accuracy achieved for each number of vehicles V and local epochs N_v. This column shows a slight performance degradation when more vehicles take part in federated training. This is due to the limited number of training samples available to each vehicle. Although we distribute the samples among vehicles uniformly at random, the local vehicle datasets are still slightly skewed due to the limited number of training samples (e.g., 11K samples available in s008). This leads to the catastrophic forgetting phenomenon [31], [32], where increasing N_v tends to overfit to local datasets, which may not efficiently represent the true distribution across the cell. This can be mitigated utilizing ideas similar to [31], [32], and is a direction for future research.

Finally, in Table III, we present the performance of our proposed federated beam selection scheme when initialized from an offline-trained model. We have considered two scenarios as below:

TABLE II: Performance of federated beam selection when initialized from an untrained global model.

V	N_v	$(N_a)^{0.88}$	$(O_{DL})^{0.88}$	$(O_{UL})^{0.88}$	Top-10 Acc.
5	1	19	$19\|\theta\|$	$95\|\theta\|$	90.12%
	2	13	$13\|\theta\|$	$65\|\theta\|$	90.34%
	5	10	$10\|\theta\|$	$50\|\theta\|$	89.92%
10	1	31	$31\|\theta\|$	$310\|\theta\|$	89.77%
	2	22	$22\|\theta\|$	$220\|\theta\|$	89.16%
	5	15	$15\|\theta\|$	$150\|\theta\|$	88.64%
20	1	81	$81\|\theta\|$	$1620\|\theta\|$	88.81%
	2	48	$48\|\theta\|$	$960\|\theta\|$	88.53%
	5	NA	NA	NA	87.33%

TABLE III: Performance of federated beam selection when initialized from an offline-trained model.

V	N_v	2K from s008 (Rosslyn)		2K from s007 (Beijing)	
		$(N_a)^{0.88}$	Top-10 Acc.	$(N_a)^{0.88}$	Top-10 Acc.
10	1	17	90.29%	23	89.35%
	2	10	89.86%	14	88.73%
	5	6	89.31%	9	88.19%

-The model is trained offline for 20 epochs on 2K samples taken randomly from Raymobtime s008 achieving an initial Top-10 accuracy of 81% on the test dataset s009.

-The model is trained off-line for 20 epochs on 2K samples taken randomly from Raymobtime s007 achieving an initial Top-10 accuracy of 56% on the test dataset s009.

In both cases, the offline-trained model is then used as the initial global model for federated training. For federated training, 9K samples from s008 are distributed randomly among the 10 vehicles participating in federated training.

Referring to [16], [19], s007, s008 and s009 are Raymobtime datasets from different locations (i.e., s008 and s009 are from Rosslyn, while s007 is from Beijing) at the same 60GHz carrier frequency. The purpose of the above two scenarios is not only to study the effects of using an initial model trained offline, but also to investigate how well the model can adapt if significant changes happen in the coverage area of the BS. It can be concluded from the results presented in Table III that using an offline trained NN generally reduces the required number of aggregation rounds (hence, reducing the communication overhead) as well as the number of samples required to be collected by the vehicles (here from 11K to 9K). If the dataset used for offline training is a good representative of the scattering environment, then a slight improvement in the accuracy is observed as well. But even if it is not, the federated training scheme can adapt the model to the new environment and still achieve a very good accuracy.

V. Conclusions

We have studied efficient link configuration in mmWave V2I communication networks, and considered exploiting side information in the form of LIDAR and position data in a supervised learning scheme to reduce the beam search overhead. In this letter, we first proposed LIDAR preprocessing and a convolutional NN architecture that improves the state-of-the-art classification accuracy with a significantly reduced model complexity. We have then proposed a federated training scheme that enables connected vehicles to collaboratively train a shared NN on their locally available LIDAR data. Once the NN is collaboratively trained, any vehicle entering the coverage area of the BS can employ it to reduce the beam search overhead.

REFERENCES

[1] 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, 2018.

[2] M. Brambilla, M. Nicoli, S. Savaresi, and U. Spagnolini, "Inertial sensor aided mmWave beam tracking to support cooperative autonomous driving," in *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2019, pp. 1–6.

[3] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, "Beam alignment for millimeter wave links with motion prediction of autonomous vehicles," in *Antennas, Propagation RF Technology for Transport and Autonomous Platforms 2017*, 2017, pp. 1–8.

[4] N. González-Prelcic, R. Méndez-Rial, and R. W. Heath, "Radar aided beam alignment in mmWave V2I communications supporting antenna diversity," in *2016 Information Theory and Applications Workshop (ITA)*, 2016, pp. 1–7.

[5] 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)*, 2015, pp. 2416–2424.

[6] A. Ali, N. González-Prelcic, and R. W. Heath, "Millimeter wave beam-selection using out-of-band spatial information," *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 1038–1052, 2018.

[7] M. Alrabeiah and A. Alkhateeb, "Deep learning for mmWave beam and blockage prediction using sub-6 GHz channels," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5504–5518, 2020.

[8] I. Chafaa, R. Negrel, E. V. Belmega, and M. Debbah, "Federated channel-beam mapping: from sub-6ghz to mmwave," in *IEEE WCNC 2021 Workshop: Distributed Machine Learning*, 2021.

[9] V. M. De Pinho, M. L. R. De Campos, L. U. Garcia, and D. Popescu, "Vision-aided radio: User identity match in radio and video domains using machine learning," *IEEE Access*, vol. 8, pp. 209 619–209 629, 2020.

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

[11] 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.

[12] 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.

[13] 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.

[14] *AI/ML in 5G*, accessed April 2021, https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/default.aspx.

[15] *AI/ML in 5G*, accessed April 2021, https://www.itu.int/en/ITU-T/AI/challenge/2020/Pages/results.aspx.

[16] *Raymobtime*, accessed November 2020, https://www.lasse.ufpa.br/raymobtime/.

[17] H. B. McMahan, E. Moore, D. Ramage, and S. Hampson, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.

[18] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, p. 1–10, Nov 2015. [Online]. Available: http://dx.doi.org/10.1016/j.neunet.2015.07.007

[19] 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.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," *Advances in Neural InformationProcessing Systems*, vol. 30, p. 1709–1720, 2017.

[22] J. Bernstein, J. Zhao, K. Azizzadenesheli, and A. Anandkumar, "signSGD with majority vote is communication efficient and fault tolerant," *International Conference on Learning Representations*, 2019.

[23] M. M. Amiri, D. Gündüz, S. R. Kulkarni, and H. V. Poor, "Federated learning with quantized global model updates," *arXiv preprint arXiv:2006.10672v2*, Oct 2020.

[24] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," *International Conference on Learning Representations*, 2018.

[25] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," *Advances in Neural Information Processing Systems 31*, vol. 31, p. 1305–1315, 2018.

[26] M. Mohammadi Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2155–2169, 2020.

[27] G. Zhu, Y. Du, D. Gündüz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Transactions on Wireless Communications*, vol. 20, no. 3, pp. 2120–2135, 2021.

[28] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *arXiv preprint arXiv:1808.05377v3*, April 2019.

[29] M. Wistuba, A. Rawat, and T. Pedapati, "A survey on neural architecture search," *arXiv preprint arXiv:1905.01392v2*, Jun 2019.

[30] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578v2*, Feb 2017.

[31] N. Shoham, T. Avidor, A. Keren, N. Israel, D. Benditkis, L. Mor-Yosef, and I. Zeitak, "Overcoming forgetting in federated learning on non-iid data," *arXiv preprint arXiv:1910.07796v1*, Oct 2019.

[32] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582v1*, Jun 2018.