META-LEARNING FOR IMAGE-GUIDED MILLIMETER-WAVE BEAM SELECTION IN UNSEEN ENVIRONMENTS

Jerry Gu[⋆]*, Liam Collins*[†]*, Debashri Roy*[⋆]*, Aryan Mokhtari*[†]*, Sanjay Shakkottai*[†]*,*
Kaushik R. Chowdhury[⋆⋆]

[⋆] Northeastern University, Boston, MA, USA, 02115
[†] The University of Texas at Austin, Austin, TX, USA, 78712

ABSTRACT

The use of alternate modalities, like images, for fast beamforming in the millimeter wave (mmWave)-band is being proposed to ensure high bandwidth connectivity in vehicular scenarios typically seen in the context of autonomous cars. Considering the dynamic deployment conditions, a car may encounter new environments which were not explicitly included in an apriori training dataset. In this paper, we propose to use the Model-Agnostic Meta-Learning (MAML) framework on the image data of the mmWave vehicle-to-infrastructure beam selection FLASH dataset, to overcome the generalization issues of a pre-trained model in unseen non-line-of-sight (NLOS) connectivity environments. MAML has additional advantages over traditional deep-learning techniques: (i) it uses a fraction of the data which, in turn, simplifies data collection and storage, and (ii) it results in equal or higher accuracy in optimal beam selection compared to the case when the new environment dataset is fully available during initial training. We show that our MAML implementation improves test accuracy of beam selection by up to 86% with fine-tuning when encountering an unseen NLOS environment compared to conventional supervised learning.

***Index Terms*—** Meta-learning, Non-RF Data, mmWave, Beam Selection

1. INTRODUCTION

Autonomous vehicles are poised to revolutionize transportation in the future. One direction towards realizing this vision involves relaying massive volumes of locally collected sensor data from the vehicle to a central cloud for optimizing routes and obstacle avoidance, in the order of 10 Gbps. Transmission in the millimeter-wave (mmWave) band makes such high data rates possible [1], though beamforming is required to overcome the high path loss in this band by channeling the radio frequency (RF) energy in narrow spatial lobes. We have previously used machine learning (ML) over contextual information from the environment, captured via images from a vehicle-mounted camera to enable beamforming faster than the standards-defined brute force approach. In this paper, we tackle the key problem of enabling our image-guided beamforming approach to work in environments not seen during training, while minimizing the costly overhead of new data collection.

• **Challenge 1: Overhead of Experimental Data Collection:** Our baseline approach for image-guided beam selection in [2] required extensive experiments for data collection using an autonomous car, mounted with a Talon AD7200 802.11ad mmWave router and GoPro HERO4 camera with a field-of-view of 130 degrees operating at a sampling rate of 30 frames per second. To collect this dataset, we drove the vehicle in an alley that is flanked on both sides by buildings to emulate typical urban driving conditions, collecting received signal strength indicator measurements (RSSI) captured by the router and multimodal data from the non-RF sensors in 1-30 Hz intervals. The resulting 20 GB dataset [3] is collected with time-synchronized sensor samples and RSSI measurements per mmWave beam. We demonstrate high accuracy in prediction of the best beam (defined as having the highest RSSI measurement) when we train and test over *all* proposed line of sight (LOS) and non-LOS (NLOS) conditions.

*The authors gratefully acknowledge the funding from the US National Science Foundation (grant CNS-2112471) and Northeastern University's Field Robotics Lab.

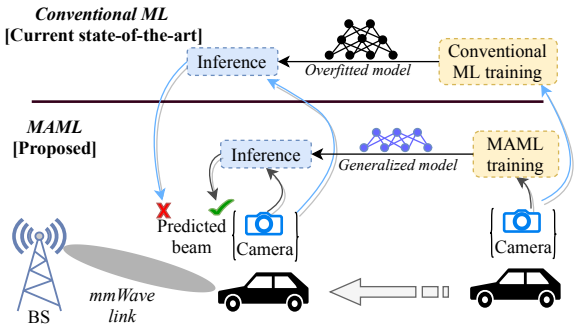

Fig. 1: State-of-the-art (top) versus proposed ML-based solution (bottom). In the former, the model is tuned to give best performance on specific scenario during training; thus, it fails while predicting for unseen scenarios which it has not been trained on. In latter, the MAML-based training generates a generalized model to perform optimally in any unseen scenarios.

• **Challenge 2: Overhead and Accuracy of Learning:** Even if experimental data can be collected in the field, the training process may be lengthy due to digital processing constraints on millions of parameters. Additionally, the time needed to relay the data to the computational cloud and complete the training is proportional to the data volume. Furthermore, as shown by recent investigations, simply using more data is not guaranteed to result in a better performance of the model due to the possibility of over-fitting [4, 5]. Specifically for the aforementioned beam selection scenario in new NLOS conditions, we desire to be parsimonious with the data used for additional training in the new environment.

• **Solution and Approach:** We propose a novel meta-learning-based solution to address Challenges 1 and 2 on a non-RF image dataset. As shown in Fig. 1, we adapt the Model-Agnostic Meta-Learning (MAML) [6] framework for beam selection. Our approach uses available training data on a number of *seen* scenarios to find an *adaptive* model that performs well after fine-tuning on new or *unseen* scenarios with only a small amount of data at test time, given that the

scenarios fall under a common distribution of tasks.

2. RELATED WORKS

It is well known that the standardized exhaustive search process for beam selection at mmWave frequencies is slow and inefficient [7]. In recent years, applications of ML have gained popularity over conventional techniques for beam selection, reducing the range of possible optimal beam sectors and leveraging environmental information to speed up the selection process. Examples that have drastically reduced beam selection time within the mmWave domain include [8, 9, 10, 11, 12]. However, models learned by these techniques fail to generalize well when they are trained in environments with very few training samples. In that regard, meta-learning-based approaches have the potential to allow ML models to quickly adapt to new scenarios and minimize the impact of generalization errors. Meta-learning, or "learning to learn" [13], aims to extract information from a set of observed tasks that enables fast adaptation to new tasks, with only a small amount of samples and computation available for adaptation. Meta-learning approaches include learning an initialization for gradient-based learning algorithms [6, 14, 15, 16, 17, 18], finding a low-dimensional subspace in which to fine-tune model parameters [19, 18], and learning a metric space containing class-specific prototypical representations [20]. To our knowledge, this paper takes the first step in applying meta-learning to a mmWave beam selection scheme using a real dataset in which non-RF camera images are used to guide inferences, and new samples may be collected and processed for inference in real-time.

3. BEAM SELECTION IN UNSEEN SCENARIOS AND PROPOSED FRAMEWORK

3.1. The Problem: Beam Selection in Changing Scenarios

Our setting involves a receiver (Rx) on a vehicle, which is selecting a beam pair with a base-station (BS) transmitter (Tx) in order to establish a link with high signal strength.

Traditional Beam Selection: Traditional beam selection leverages phased antenna arrays at the Tx of the BS and Rx at the vehicle. Each array has a pre-defined codebook given by $C_{Tx} = \{t_1, \ldots, t_{P_i}\}$, $C_{Rx} = \{r_1, \ldots, r_{Q_i}\}$ consisting of P_i and Q_i elements for i^{th} scenario, respectively, where each scenario corresponds to the vehicle travelling at a different speed and/or with a different obstacle between it and the base station. These $P_i + Q_i$ probe frames are transmitted for beam initialization, and the beam with the maximum signal strength across sectors is chosen as optimal. In particular, the optimal beam at Tx is:

$$t_i^* = \underset{1 \le p_i \le P_i}{\arg\max}\, y_{t_{p_i}}, \tag{1}$$

where $y_{t_{p_i}}$ is the strength of the received signal when the transmitter uses beam t_{p_i} for the i^{th} scenario. While this method is effective at finding the optimal beam, exhaustively transmitting a linearly-scaling $P_i + Q_i$ probe frames is slow and impractical within a vehicle-to-everything (V2X) network, particularly when considering a large number of possible beams and transmission time [21]. By the time the optimal beam is found, the vehicle may have moved into a new environment where either a different number of beams are available or the optimal beam is no longer available, requiring the selection process to be reinitialized.

ML-based Beam Selection: To avoid costly exhaustive searching in traditional beam selection, we follow state-of-the-art techniques [2, 22] that employ ML models to predict the best beam based on non-RF data. In this work, we focus on using image data to predict the best beam, motivated by the fact that each vehicle is equipped with an active camera and large quantities of image data are available. Our aim is to use a large set of collected image data to train a model in an offline manner that predicts the optimal beam from an image taken by the vehicle's camera, but our framework can be easily extended to other non-RF modalities as well.

More formally, prior to model deployment, we access a training set of labeled images from η different scenarios. Training data corresponding to the i^{th} scenario is given by $\mathcal{S}_i := \{(X_{i,j}, y_{i,j})\}_{j \in n_i}$, where each $X_{i,j} \in \mathbb{R}^d$ is an image and $Y_{i,j} \in \{0,1\}^{P_i}$ is its corresponding label, and n_i is the number of samples from the i^{th} scenario. The learning model is a function $f_\theta : \mathbb{R}^d \mapsto \mathbb{R}^{P_i}$, parameterized by $\theta \in \mathbb{R}^D$, e.g., f_θ may be a neural network with weights θ. The empirical loss of the model parameters θ on a dataset $\mathcal{S}_i$ is defined as $\mathcal{L}(\theta; \mathcal{S}_i) := \frac{1}{n_i} \sum_{j=1}^{n_i} [\ell(f_\theta(X_{i,j}), y_{i,j})]$, where $\ell : \mathbb{R}^{P_i} \times \{0,1\}^{P_i} \to \mathbb{R}^+$ is a cross-entropy loss function measuring the discrepancy between predicted and true labels that will be used in all experiments.

The standard ML training approach is to find a model that minimizes the average loss across all of the training samples, namely, empirical risk minimization (ERM). Specifically, ERM solves $\min_{\theta \in \mathbb{R}^D} L(\theta) := \frac{1}{N} \sum_{i=1}^{\eta} n_i \mathcal{L}(\theta, \mathcal{S}_i)$, where $N = \sum_{i=1}^{\eta} n_i$. One can run a variety of easy-to-implement gradient-based algorithms to optimize the above objective–for instance, stochastic gradient descent (SGD). While this approach is natural for finding high-performing models on training scenarios, it is not well-suited to find models that can adapt to new scenarios encountered when deployed, as we show in Section 4.

Beam Selection in Unseen Scenarios: We are interested in finding models $\hat{\theta}$ which, after they are deployed, can quickly adapt to scenarios $\mathcal{S}_{\eta+1}, \ldots, \mathcal{S}_{\eta+\upsilon}$ not seen during training. In practical applications such as V2X networks, the model does not have enough data or computational budget to perform full supervised learning in the new scenario; rather, it is only provided with a few labeled samples and must yield an adapted model within a matter of seconds or less, as a vehicle may only be within communication range of a BS for a few seconds.

We consider each of these adaptation opportunities as a "task". That is, a task consists of a small number of support samples that can be used for adapting the model, along with target samples for evaluating the adapted model. Each task has data that is a subset of the dataset for a particular scenario. Specifically, the k^{th} task from scenario i is defined by the pair of datasets $(\mathcal{T}_{i,k}^{sup}, \mathcal{T}_{i,k}^{tar})$, where $\mathcal{T}_{i,k}^{sup}$ contains the support samples, $\mathcal{T}_{i,k}^{tar}$ contains the target samples, $\mathcal{T}_{i,k}^{sup} \cup \mathcal{T}_{i,k}^{tar} \subseteq \mathcal{S}_i$ and $\mathcal{T}_{i,k}^{sup} \cap \mathcal{T}_{i,k}^{tar} = \emptyset$. We let $m_1 := |\mathcal{T}_{i,k}^{sup}|$ and $m_2 := |\mathcal{T}_{i,k}^{tar}|$, for all tasks i, k, and let M_i denote the number of tasks for scenario i.

We suppose that the task-specific adaptation procedure is τ steps of gradient descent (GD) with step size α using the support samples in the task's support set, where τ is small. Let $\hat{\theta}_{i,k} := \text{GD}(\hat{\theta}, \mathcal{T}_{i,k}^{sup}, \alpha, \tau)$ denote the result of this adaptation procedure starting from $\hat{\theta}$. Ultimately, we aim to find a $\hat{\theta}$ such that the loss of $\hat{\theta}_{i,k}$ is small on average across tasks from unseen scenarios, i.e., our performance metric is: $L_{\text{adapt}}^{\text{test}}(\hat{\theta}) := \frac{1}{\upsilon} \sum_{i=\eta+1}^{\eta+\upsilon} \sum_{k=1}^{M_i} \mathcal{L}(\hat{\theta}_{i,k}; \mathcal{T}_{i,k}^{tar})$. As mentioned previously, models found by the standard ERM are not well-suited to perform well on the adaptive metric $L_{\text{adapt}}^{\text{test}}(\cdot)$ due to overfitting during training. Unlike ERM, we leverage the seen scenarios to train for adaptivity, described next.

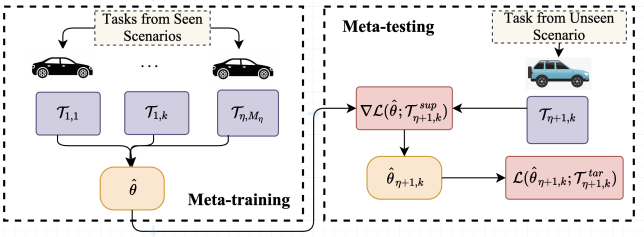

Fig. 2: Proposed MAML-based framework for adapting to unseen scenarios for beam selection. $\hat{\theta}$ is the model after meta-training, and $\hat{\theta}_{\eta+1,k}$ is generated after fine-tuning during meta-testing.

3.2. Proposed Solution: MAML for Beam Selection in Unseen Scenarios

In order to find models that perform well after adaptation, i.e., achieve a small $L^{adapt}(\cdot)$, we adopt the MAML [6] approach. MAML aims to find an adaptable initialization for task-specific SGD in multi-task settings. To do so, MAML executes an episodic training procedure, referred to as meta-training, in which each episode consists of first adapting the current initialization to the corresponding task, then improving the initialization based on the performance of the adapted model on the same task. In particular, MAML aims to solve the following objective in our setting:

$$\min_{\theta\in\mathbb{R}^D} L_{\text{adapt}}^{\text{train}} := \frac{1}{\eta}\sum_{i=1}^{\eta}\sum_{k=1}^{M_i}\mathcal{L}(\theta_{i,k};\mathcal{T}_i^{tar}) \tag{2}$$

where $\theta_{i,k} := GD(\theta, \mathcal{T}_{i,k}^{sup}, \alpha, \tau)$. In words, we aim to find an initial model $\hat{\theta}$ that performs well after τ GD steps using the samples $\mathcal{T}_i^{sup}$, on average across all scenarios indexed by i. To solve (2), we execute the MAML algorithm, which is equivalent to performing SGD on (2). This framework is displayed in Fig. 2. Note that the MAML training objective $L_{\text{adapt}}^{\text{train}}$ is the analogue of $L_{\text{adapt}}^{\text{test}}$ on the training data. Indeed, our evaluation procedure on tasks from unseen scenarios exactly corresponds to what MAML refers to as the meta-testing phase.

4. EXPERIMENTS

4.1. Dataset

We validate our proposed framework on the publicly available FLASH dataset for multimodal beamforming [3]. The FLASH dataset studies a vehicle-to-infrastructure scenario at the 60 GHz mmWave band and includes synchronized sensor data from on-board GPS, a GoPro Hero4 camera, and two Velodyne VLP-16 LiDARs, along with the received signal strength indicator for all beams recorded by the Talon AD7200 mmWave radio [7]. The latitude and longitude of the vehicle and side view of the vehicle are recorded by the on-board GPS and RGB camera with shape (90, 160, 3), respectively, as the vehicle passes by a static BS. For this paper, we use the ∼32000 camera images from FLASH dataset for guided beam selection.

A variety of images, or *samples*, are collected in LOS and NLOS conditions with various obstacles intended to comprehensively represent many V2X environments. These samples are subsequently used to train a model that selects the best mmWave beam sector. Specifically, these samples, collected in trials or *episodes*, are organized by broad *categories* with specific *scenarios* per category. Since the FLASH dataset consists of synchronized sensor and RF ground truth data, each collected sample is labeled with the optimal beam sector at the time of recording.

Cat.	Seen/ Unseen	Featuring	Scenario
1	Seen	LOS	10 mph, same lane
	Seen	LOS	15 mph, same lane
	Seen	LOS	20 mph, same lane
	Seen	LOS	10 mph, opp. lane
	Seen	LOS	15 mph, opp. lane
	Seen	LOS	20 mph, opp. lane
2	Seen	Pedestrian	15 mph, standing
	Seen	Pedestrian	15 mph, walking L → R
	Seen	Pedestrian	15 mph, walking R → L
	Seen	Pedestrian	15 mph, walking F → B
	Seen	Pedestrian	15 mph, walking B → F
3	Seen	Static Car	15 mph, on R
	Seen	Static Car	20 mph, on R
	Seen	Static Car	15 mph, in F
	Seen	Static Car	20 mph, in F
	Seen	Static Car	15 mph, on L
	Seen	Static Car	20 mph, on L
4	Unseen	Moving Car	10/15 mph, same direction
	Unseen	Moving Car	15/20 mph, same direction
	Unseen	Moving Car	15/15 mph, opp. direction
	Unseen	Moving Car	20/20 mph, opp. direction

Table 1: Summary of different used scenarios. L, R, F, and B stand for left, right, front, and back, respectively. Samples for Cat. 2-4 are taken with the Rx vehicle in the lane opposite to the BS, while any obstacles occupy the same lane as the BS. For Cat. 4, the speed of the Rx vehicle is given first in each scenario, while the speed of any moving car obstacle, when present, is given second.

4.2. Seen and Unseen Scenarios

In our experiments, scenarios are either *seen* or *unseen*, and the seen scenarios are used for training, while the unseen scenarios are used for testing. In particular, scenarios from Cat. 1-3 are seen and samples from Cat. 4 are unseen. Then, we configure the scenarios based on the speed of the vehicle and the available scenarios within each category with the sectors 1-31 and 61-63 [21] as the classes. Since there are 34 possible sectors, each task is a 34-way classification, and task data is drawn uniformly from the datasets for each scenario. The scenarios are shown in Tab. 1, with corresponding categories, obstacles, and descriptions based on vehicle speeds and scenarios; details about these scenarios are given in [2].

4.3. Experiment Setup

We conduct experiments to explore the advantages of meta-learning based training. For all experiments, we set the batch size to 5 tasks sampled per epoch for 100 epochs and use the Adam optimizer [23] with a learning rates of 0.001 for ERM and MAML, and adaptation learning rate $\alpha = 0.01$. The learning model is VGGNet [24]–a convolutional neural network that is widely used for image classification. Note that in the FLASH dataset, each class may contain up to ∼1500 samples, depending on the optimal beam per scenario in each category.

4.3.1. ERM

As discussed in Section 3.1, a natural ML-based approach to beam selection is ERM, which tries to find a model that minimizes the average loss on the seen scenarios. For a fair comparison with MAML, we sample data in the form of tasks. However, we use all of the task

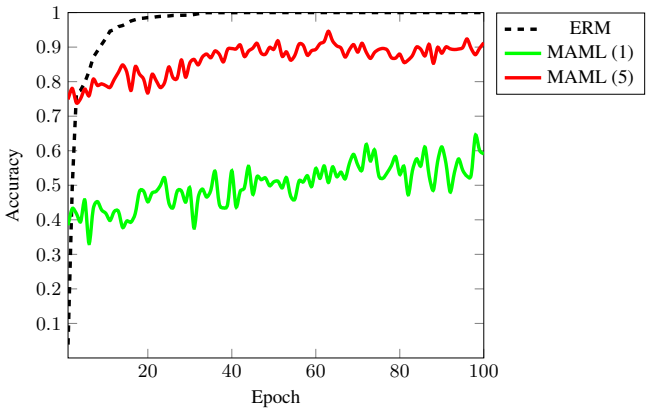

Fig. 3: Training accuracies per epoch for different experiment setups with 1 FT step. Additional FT steps do not impact training. The number of task-specific adaptation steps τ that MAML uses during training is given in parentheses.

data for a single gradient step, rather than splitting task samples into support and target sets. We use 120 samples per class per task for a total of $m := m_1 + m_2 = 4080$ samples per task, as there are 34 classes per task.

4.3.2. MAML

To implement MAML, we use 20 support samples and 100 target samples from each class per task. Since each task has 34 classes, this means that $m_1 = 680$ and $m_2 = 3400$ for all tasks. Note that ERM and MAML use the same number of samples per task for fair comparison. We experiment with two versions of MAML; (i) *MAML (1)*, which makes $\tau = 1$ gradient step for model adaptation in the meta-training phase, and (ii) *MAML (5)*, which makes $\tau = 5$ steps. For run-time efficiency, we take the common approach of dropping second-order derivatives in the MAML update, meaning the version of MAML we run is First-Order MAML [6].

4.3.3. Evaluation

For evaluation, the trained models are fine-tuned on new tasks from the 'unseen' scenarios using either $\tau = 1$ or $\tau = 5$ fine-tuning (FT) steps with 20 samples per class per task, for a total of 680 samples per task. Then, the accuracy of the fine-tuned models is computed using 100 new samples per class from the same task. The average and standard deviation of the test accuracies across test tasks is reported in the following section.

5. EXPERIMENTAL RESULTS

We perform all experiments on a NVIDIA Tesla A100 GPU machine. We implement MAML and ERM in Pytorch by adapting the codebase in [25]. For all experiments, average accuracies on training tasks per epoch are shown in Fig. 3 while average accuracies and standard deviations on testing tasks are shown in Fig. 4.

Comparison between ERM and MAML: Fig. 3 shows that ERM quickly learns the mapping from inputs images to the optimal beam sector among images from the training scenarios, achieving nearly 100% training accuracy. However, its solution does not generalize to tasks from unseen scenarios, as it obtains only 11.76% average accuracy on tasks from unseen scenarios for both 1 and 5 FT steps due to overfitting, as shown in Fig. 4. In comparison, MAML (1) and MAML (5) obtain smaller training accuracies yet generalize much better during testing, collectively achieving an average test accuracy of 66.70% and 97.83% across all test cases and FT steps.

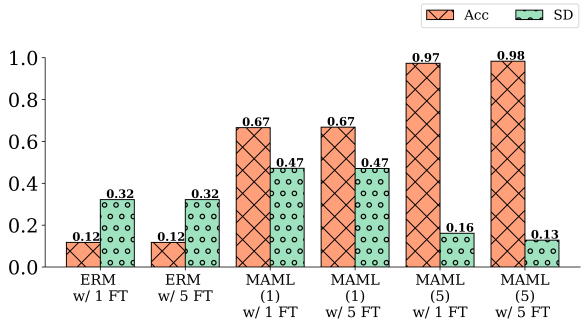

Fig. 4: Comparison of testing accuracies and standard deviations between different settings of ERM and MAML with different number of FT steps.

Observation 1. *We observe that MAML training significantly boosts the testing accuracy while encountering the unseen scenarios versus the standard ERM training method.*

Impact of the Number of Adaptation Steps During Training: We also explore the effects of increasing the number of task-specific adaptation steps between MAML (1) and MAML (5). As shown in Fig. 3 and Fig. 4, MAML (5) achieves a larger test accuracy of 97.33% after 1 step of FT than MAML (1) (66.58%), as well as a smaller standard deviation in test accuracy across test tasks. The results are similar for 5 steps of FT.

Observation 2. *We observe that more task-specific adaptation steps during training yields higher efficacy, drastically increasing testing accuracies while lowering testing standard deviation across tasks.*

Impact of Number of Fine-Tuning Steps: Finally, we explore the effects of additional FT on each setup. For ERM, as seen in Fig. 4, FT does not change test accuracy or standard deviation. In MAML (1) with five FT steps (w/ 5 FT), testing accuracy increases by 0.24%, while standard deviation decreases by 0.08% when compared to MAML (1) w/ 1 FT. In MAML (5), 5 steps of FT boosts testing accuracy by 1.00% and decreases the standard deviation by 3.29% relative to 1 step of FT.

Observation 3. *Five FT steps generally increase testing accuracies and decreases standard deviation by a small margin compared to one FT step.*

6. CONCLUSION

In this paper, we demonstrate how a ML algorithm can adapt to new, unseen scenarios by implementing meta-learning on images from the FLASH dataset for mmWave beam selection in V2X networks. In our experiments, we show how using the MAML meta-training framework results in a model that can generalize to new tasks with only 20 FT samples per class, unlike models learned using the standard ML training procedure, ERM. Depending on the number of task-specific adaptation steps being used during meta-training and the amount of FT during the testing phase, final testing accuracy increases by up to 86% compared to the standard ERM. Our future work involves extending the MAML framework to include an additional sensor modality during training and test, e.g., the LiDAR data in FLASH in combination with the camera images used in this paper, as well as implementing a task-robust version of MAML [26] to reduce the standard deviation in performance across test tasks.

7. REFERENCES

[1] Qualcomm, "5G MmWave," 2022. [Online]. Available: https://www.qualcomm.com/research/5g/5g-nr/mmwave

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

[3] "Multimodal Fusion for NextG V2X Communications," 2022. [Online]. Available: https://genesys-lab.org/multimodal-fusion-nextg-v2x-communications

[4] C. F. Santos and J. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Computing Surveys*, vol. 54, no. 10s, p. 1–25, 2022.

[5] X. Ying, "An overview of overfitting and its solutions," *Journal of Physics: Conference Series*, vol. 1168, p. 022022, 2019.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[7] 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*, 2017.

[8] D. Last, P. Thomas, S. Hiscocks, J. Barr, D. Kirkland, M. Rashid, S. B. Li, and L. Vladimirov, "Stone soup: Announcement of beta release of an open-source framework for tracking and state estimation," in *Signal Processing, Sensor/Information Fusion, and Target Recognition XXVIII*, vol. 11018. International Society for Optics and Photonics, 2019, p. 1101807.

[9] J. Jia and H. Duan, "Automatic target recognition system for unmanned aerial vehicle via backpropagation artificial neural network," *Aircraft Engineering and Aerospace Technology*, 2017.

[10] 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.

[11] 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*, 2020, pp. 1–5.

[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] S. Thrun and L. Pratt, "Learning to learn: Introduction and overview," in *Learning to learn*. Springer, 1998, pp. 3–17.

[14] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," *CoRR*, 2018.

[15] M. Khodak, M.-F. F. Balcan, and A. S. Talwalkar, "Adaptive gradient-based meta-learning methods," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[16] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations*, 2016.

[17] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[18] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," *International Conference on Learning Representations*, 2019.

[19] Y. Lee and S. Choi, "Gradient-based meta-learning with learned layerwise metric and subspace," in *International Conference on Machine Learning*, 2018, pp. 2927–2936.

[20] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] D. Steinmetzer, D. Wegemer, and M. Hollick, "Talon Tools: The Framework for Practical IEEE 802.11ad Research," 2018. [Online]. Available: https://seemoo.de/talon-tools/

[22] B. Salehi, G. R. Muns, D. Roy, Z. Wang, T. Jian, J. G. Dy, S. Ioannidis, and K. R. Chowdhury, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *IEEE Transactions on Vehicular Technology*, 2022.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] A. Antoniou, H. Edwards, and A. Storkey, "How to Train your MAML in Pytorch." [Online]. Available: https://github.com/AntreasAntoniou/HowToTrainYourMAMLPytorch

[26] L. Collins, A. Mokhtari, and S. Shakkottai, "Task-robust model-agnostic meta-learning," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 18 860–18 871.