

# TUNE: Transfer Learning in Unseen Environments for V2X mmWave Beam Selection

Jerry Gu, Batool Salehi, Snehal Pimple, Debashri Roy, Kaushik R. Chowdhury  
*Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA, 02115*  
 {gu.je, salehikouei.b, pimple.sne}@northeastern.edu, {droy, krc}@ece.neu.edu

**Abstract**—The use of non-RF data can potentially speed up millimeter wave-band sector-steering in vehicular mobility scenarios by gaining contextual knowledge of the environment. While several works have demonstrated the benefits of this approach, especially applying machine learning models on inputs from LiDAR and image sensors, adapting such models in ‘unseen’ environments remains an open problem. State-of-the-art techniques generally use a single, pre-trained model for all different scenarios, which assumes that the network has ‘seen’ representative examples of all future scenarios. In this paper, we propose the TUNE framework, which solves this problem by: (a) transfer learning (TL) for better performance with similar convergence times in comparison to non-TL-generated model testing, (b) utilizing statistical properties to select the best-suited starting ‘seen’ scenario (and by extension the model trained for it), and (c) a refinement of the transfer learning framework by dynamically selecting the most pertinent layers for retaining, thus reducing the overhead compared to fully retraining a model. We validate TUNE on publicly available synthetic and real-world datasets for mmWave beam selection for V2X communication, revealing that TUNE generally outperforms non-TL methods in a variety of tasks where a different number of beams is available between the training and testing environments.

**Index Terms**—Transfer Learning, Multimodal Data, mmWave, Beam Selection

## I. INTRODUCTION

There is a tremendous spectrum crunch today in the sub-6 GHz frequency range. To address this shortage, operators are exploring deployment in millimeter wave (mmWave)-band frequencies such as the 28 GHz band [1] and the 57-71 GHz, which are less occupied and permit wide-band channel use resulting in data rates of over 10 Gbps [2]. The emerging vehicle-to-everything (V2X) communication paradigm relies on such links, in which vehicles may communicate a large amount of information to other vehicles and participating infrastructures for purposes such as vehicular control, road safety and efficient route planning. Most vehicles today already have a variety of non-radio frequency (RF)-based sensors installed. We believe that not only can mmWave links help in relaying all this sensor data for off-site analysis, but these sensors themselves can play a symbiotic role by aiding the formation of directional links.

**Multimodal sensing data for mmWave beam selection:** Combining different types of multimodal data and using machine learning (ML) on these datasets may boost system-wide performance in decision-making tasks. In some scenarios, non-RF data, such as data in the form of camera images, GPS coordinates, light detection and ranging (LiDAR) pointclouds,

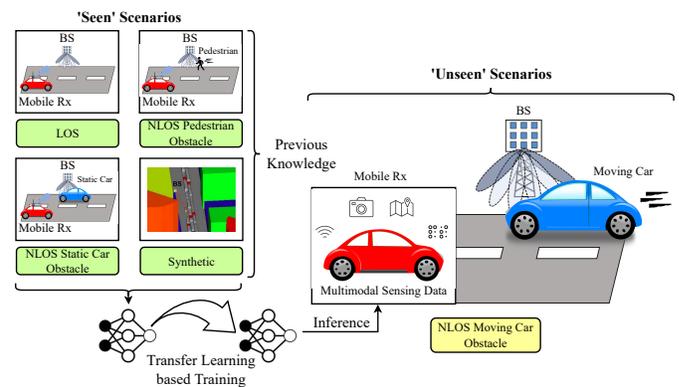


Fig. 1: A schematic of the TUNE framework.

radar signals, infrared signals, and acoustic signals, to name a few, have been used to provide additional context in decision making not otherwise observable by the use of RF data alone. Of particular relevance to this paper is the work on using non-RF data for mitigating the characteristic signal attenuation that comes with transmission at mmWave frequencies [3], [4], [5]. These multimodal datasets have been used to address channel estimation, interference detection, and beam selection in MIMO systems.

**Challenges of mmWave beam selection in V2X communication:** V2X communication is often dynamic, particularly in urban environments where the surroundings rapidly change. Obstacles such as buildings, pedestrians, flora, other vehicles, and even the weather all influence the connectivity between a base station (BS), or transmitter (Tx), and a mobile receiver (RX) at mmWave frequencies. In order for an end-to-end V2X system to function, the system must be able to rapidly adapt from line-of-sight (LOS) to non-LOS (NLOS) connections between the BS and the Rx, and also be resilient to numerous types and numbers of obstacles. This is complicated by the fact that some beams are occluded by the obstacles in a NLOS scenario, and both the optimal beam from a previous environment and the same number of available beams may not be present in a new environment.

**Transferring previously-learned knowledge:** Recently, ML has been gateway tool for overcoming many challenges in the wireless communication domain. Exploitation of multimodal sensing data to reduce aspects of mmWave communication has been widely researched [6], [7], [8], [9]. However, overcoming the challenge of *adaptive learning* in V2X communication

is yet an under-addressed problem in this domain. In that regard, *transfer learning* (TL) can be applied to leverage the gained knowledge from one scenario, typically known as the source domain, to a different but related scenario, known as the target domain, for faster or more efficient learning while accommodating rapid changes in the propagation environment, such as changing connections in a V2X environment.

**Problem definition:** In this paper, we assume that an autonomous vehicle equipped with GPS, camera and LiDAR sensors is seeking to communicate with a roadside mmWave BS. The GPS, camera and LiDAR sensors capture different aspects of the environment in a specific scenario and feed this information to a trained model to predict the optimum beam to establish a directional link with the BS. For a scenario in which the vehicle does not have a trained model, the vehicle needs to quickly learn from previously observed knowledge. We call the previously observed scenarios *seen* scenarios, and new scenarios *unseen* scenarios. To that end, there are a few significant questions, related to unseen scenarios: First, if the vehicle arrives in an unseen environment where the BS has a different number of beams than previously observed, how quickly can the vehicle transfer the knowledge from seen scenarios to infer the best possible beam? Second, if the vehicle has trained models from multiple seen scenarios, which model do we use as the starting point for knowledge transfer? Finally, how does the vehicle determine which layers to freeze or keep the same during TL between models while maintaining inference performance?

**Contributions of this paper:** We propose the TUNE framework that has the following main contributions:

- We propose a TL-based approach for adapting the common situation in which a BS in a new scenario has a different number of beams than those from previously seen scenarios. Moreover, TUNE can adapt to the scenarios recorded by these categories and can adapt to the new number of beams that are available.
- We propose the use of a statistical dependency metric called the Hilbert-Schmidt Independence Criteria (HSIC) to determine the best initialization point and learning path for efficient performance when using the previously collected data from multiple seen scenarios.
- We describe the several TL strategies that we use and list the benefits of using each strategy. We orchestrate a TL strategy that freezes each layer in the ML architecture dynamically on a case-by-case basis over the dataset to decrease the training time.
- To validate TUNE, we use an existing real-world multimodal dataset for beam selection in vehicular scenarios, FLASH [6], wherein the multiple categories of the dataset each have a different number of beams available for beam selection. We further show the effectiveness of TL by training our framework on a completely synthetic multimodal dataset used in mmWave beam selection (Raymobtime [10]) and testing the trained framework on the real-world dataset (FLASH).

## II. RELATED WORKS

### A. Reducing mmWave Beam Selection Overhead with Non-RF Data using ML

There are numerous works proposed for optimizing mmWave-band communication using ML-aided approaches by leveraging non-RF data. In [11], Mashhadi *et al.* use a federated LiDAR-aided approach to reduce beam selection overhead among locally-connected vehicles. Meanwhile, Zecchin *et al.* use the added information of the receiver (Rx) position from LiDAR data in a convolutional neural network-based framework for more efficient beam selection [12]. Xu *et al.* instead use image-based three-dimensional reconstructions of the environment to increase beam selection accuracy [13]. Although these papers demonstrate promising results, they (a) are confined to either only one or two non-RF sensor modalities, (b) do not use the data collected or models generated from previously seen scenarios to guide beam selection, and (c) have only been tested on simulation datasets, namely, the benchmark vehicle-to-infrastructure (V2I) simulation dataset Raymobtime [10]. In this paper, we use both real-world and simulation data in multiple TL schemes and data from three modalities collecting information from a number of different environments.

### B. Transfer Learning

TL itself is also an increasingly popular technique used to improve ML performance in several RF-based applications. In [14], Rezaie *et al.* leverages prior knowledge and model layer freezing from a source environment and antenna configuration to aid in beam selection using instanced simulated indoor environments. Conversely, Askarizadeh *et al.* approach wireless channel selection with TL in [15] by attributing computation and communication costs and rewards to each TL action in an economic notion to optimize their learning framework. Naturally, these sources may benefit from some of the aspects presented in this framework, such as multimodal data for increased TL efficiency, and none of these strategies consider (a) changes to the total number of supported beams at the base station (BS) as the environment, (b) utilizing the contextual relevance from the most similar previously-observed scenario from a multitude of options for efficient TL, or (c) validating techniques on both simulated and real-world datasets.

## III. TUNE SYSTEM ARCHITECTURE

### A. Problem Statement

Consider a Tx and Rx pair equipped with phased antenna arrays with a predefined codebook defined by  $C_{Tx} = \{t_1, \dots, t_{U_i}\}$ ,  $C_{Rx} = \{r_1, \dots, r_{V_i}\}$  consisting of  $U_i$  and  $V_i$  elements for  $i^{\text{th}}$  scenario, respectively. Note that Tx and Rx are allowed to have different number of beams in different scenarios. In this case, A total of  $U_i + V_i$  probe frames must be transmitted to complete the sector level sweep and the beam that returns the maximum received signal strength is

then selected as the optimum beam. For example, the optimal beam at Tx is derived by:

$$t_i^* = \arg \max_{1 \leq u_i \leq U_i} y_{t_{u_i}} \quad (1)$$

with  $y_{t_{u_i}}$  being the observed received signal strength at the Rx side when the transmitter is configured at beam  $t_{u_i}$  for the  $i^{\text{th}}$  scenario.

While effective, this method of exhaustive search involving  $U_i + V_i$  probe frames is slow for a V2X network, particularly when considering a large variation in possible beams and transmission time [16] across different environments.

### B. Proposed Solution

TUNE solves the above problem by proposing a learning paradigm on how to intelligently and dynamically transfer the knowledge from all previously ‘seen’ scenarios to a newly ‘unseen’ scenario.

- **TL to unseen scenarios:** First, we show that by using a conventional TL approach, TUNE can adapt to the ‘unseen’ scenario that may contain both a different environment and a new BS setting (discussed in Sec. IV-B).
- **HSIC-based model selection for TL:** In the case that TUNE has the option to transfer the knowledge from multiple ‘seen’ scenarios, TUNE chooses the scenario with best HSIC dependencies for the most efficient knowledge transference (see Sec. IV-C).
- **Determination of optimal TL strategy:** After TUNE chooses the ‘seen’ scenario from which to transfer knowledge from, TUNE leverages a thresholding mechanism to choose the most pertinent layers for dynamic TL in a *case-by-case* basis, as described in Sec. IV-D.

## IV. TUNE FRAMEWORK DESIGN

The TUNE framework uses the models presented in [6] as a starting point, with modifications for TL as described in this section and Sec. VI.

### A. Notations

We consider each scenario  $E_i$  to have multimodal sensing data from location coordinates (GPS), images, and LiDAR sensors. We denote the data matrices generated from these GPS, images, and LiDAR sensors as  $X_C^{E_i}$ ,  $X_I^{E_i}$ , and  $X_L^{E_i}$  with a label matrix  $Y^{E_i}$  for all three modalities. Here, considering the dimensionality of each modality with GPS co-ordinates having latitude and longitude,  $X_C^{E_i} \in \mathbb{R}^{N_t \times 2}$ ,  $X_I^{E_i} \in \mathbb{R}^{N_t \times d_0^i \times d_1^i}$ , and  $X_L^{E_i} \in \mathbb{R}^{N_t \times d_0^i \times d_1^i \times d_2^i}$  respectively, where  $N_t$  is the number of training samples. We consider BSs in different scenarios that can have variable numbers of available beams, denoting  $S^{E_i}$  beams for scenario  $E_i$ . The label matrix is denoted as  $Y^{E_i} \in \{0, 1\}^{N_t \times S^{E_i}}$  to represent the one-hot encoding of  $S^{E_i}$  beams of scenario  $E_i$ , where the optimum beam is set to 1, and rest are set to 0 as per Eq. (1).

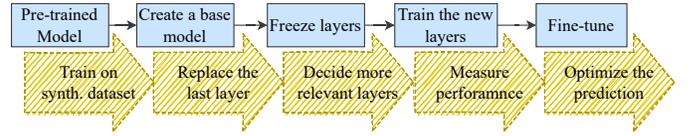


Fig. 2: A typical pipeline for transfer learning.

We define the penultimate, or second-to-last, layer fusion network which uses the  $X_C^{E_i}$ ,  $X_I^{E_i}$ , and  $X_L^{E_i}$  data matrices as:

$$\mathbf{z}_{\text{FN}}^{E_i} = \mathcal{P}_{\theta_{\text{FN}}^{E_i}}(X_C^{E_i}, X_I^{E_i}, X_L^{E_i})$$

$$\mathcal{P}_{\theta_{\text{FN}}^{E_i}}: \mathbb{R}^{2+(d_0^i+d_1^i)} + (d_1^i \times d_1^i \times d_2^i) \mapsto \mathbb{R}^{d_{\text{FN}}^{E_i}}$$

where  $\mathcal{P}_{\theta_{\text{FN}}^{E_i}}$  maps the input vector to a vector with dimension  $d_{\text{FN}}^{E_i}$ . The ultimate layer of the fusion network is presented as:

$$\mathbf{s}_{\text{FN}}^{E_i} = \sigma(\mathcal{U}_{\theta_{\text{FN}}^{E_i}}(\mathbf{z}_{\text{FN}}^{E_i})) \quad \mathcal{U}_{\theta_{\text{FN}}^{E_i}}: \mathbb{R}^{d_{\text{FN}}^{E_i}} \mapsto \mathbb{R}^{|Y^{E_i}|}$$

where  $\mathbf{s}_{\text{FN}}^{E_i}$  is the prediction score of the network and  $\mathcal{U}_{\theta_{\text{FN}}^{E_i}}$  maps the vector  $\mathbf{z}_{\text{FN}}^{E_i}$  to the one-hot encoded output. The mapping  $\sigma: \mathbb{R}^{|Y^{E_i}|} \mapsto (0, 1)^{|Y^{E_i}|}$  signifies a Softmax activation.

### B. TL for Unseen Scenarios

The generic problem of TL can be formulated as a function which maps the neural network parameters from seen scenario  $E_i$  to unseen scenario  $E_u$ . We represent the generic transfer learning problem as  $\theta_{\text{FN}}^{E_u} = \mathcal{T}(\theta_{\text{FN}}^{E_i})$ , where  $\mathcal{T}(\cdot)$  is the strategy of transferring the knowledge which will be realized.

In TUNE, we generate the prediction scores  $\mathbf{s}_{\text{FN}}^{E_u}$  of  $j^{\text{th}}$  scenario from the penultimate vector  $\mathbf{z}_{\text{FN}}^{E_i}$  of  $i^{\text{th}}$  scenario. In that case, the prediction scores for  $E_u$  scenario are formulated as:

$$\mathbf{z}_{\text{FN}}^{E_u} = \mathcal{P}_{\theta_{\text{FN}}^{E_u}}(X_C^{E_u}, X_I^{E_u}, X_L^{E_u})$$

$$\mathbf{s}_{\text{FN}}^{E_u} = \sigma(\mathcal{U}_{\theta_{\text{FN}}^{E_u}}(\mathbf{z}_{\text{FN}}^{E_u}))$$

$$\mathcal{P}_{\theta_{\text{FN}}^{E_u}}: \mathbb{R}^{2+(d_0^u+d_1^u)} \times (d_1^u \times d_1^u \times d_2^u) \mapsto \mathbb{R}^{d_{\text{FN}}^{E_u}}$$

$$\mathcal{U}_{\theta_{\text{FN}}^{E_u}}: \mathbb{R}^{d_{\text{FN}}^{E_u}} \mapsto \mathbb{R}^{|Y^{E_u}|}$$

In other words, we use the same function  $\mathcal{P}_{\theta_{\text{FN}}^{E_i}}$  for generating vector  $\mathbf{z}_{\text{FN}}^{E_i}$  of scenario  $E_i$  and pass to the new ultimate layer  $\mathcal{U}_{\theta_{\text{FN}}^{E_u}}$  which maps to a different one-hot encoded label matrix  $Y^{E_u}$ . In this case, the  $\mathcal{T}(\cdot)$  is realized by the newly added ultimate layer  $\mathcal{U}_{\theta_{\text{FN}}^{E_u}}$  for unseen scenario  $E_u$ . Following the proposed concept, a typical TL pipeline is shown in Fig. 2, where we obtain the vector  $\mathbf{z}_{\text{FN}}^{E_i}$  after the ‘freeze layer’ block. The  $\mathcal{U}_{\theta_{\text{FN}}^{E_u}}$  represents the ‘train the new layers’ block.

### C. Hilbert-Schmidt Independence Criterion-based Model Selection for TL

We assume that when TUNE encounters an unseen scenario  $E_u$ , it has previously observed  $\mathcal{N}$  number of seen scenarios  $E_i$  with  $1 \leq i \leq \mathcal{N}$ , and models  $\mathcal{P}_{\theta_{\text{FN}}^{E_i}}$ , which generate vectors  $\mathbf{z}_{\text{FN}}^{E_i}$ . To determine which model to select for the TL approach

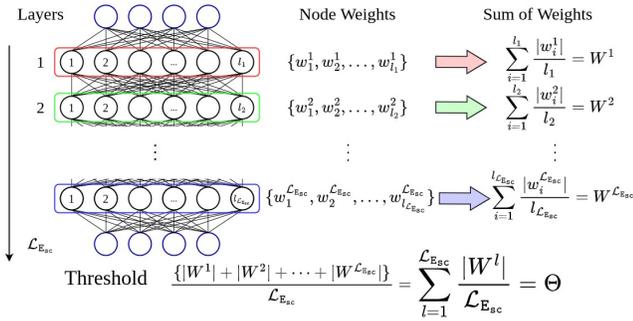


Fig. 3: Calculation of the threshold used to decide which layers to freeze or retrain in an example network. We freeze all layers  $l$  such that the normalized sum of weights from that layer  $W^l$  is greater than or equal to the calculated freezing threshold  $\Theta$ .

discussed in Sec. IV-B, we use HSIC, a statistical dependency metric used to measure non-linear dependency between two random variables, first introduced by Gretton *et al.* [17], to measure the relevance of the generated features  $\mathbf{z}_{FN}^{E_u}$  for  $E_u$  with  $\mathbf{z}_{FN}^{E_j}$  for  $E_j$ .

In TUNE, we calculate HSIC with an empirical approximation  $\widehat{\text{HSIC}} \sim \text{HSIC}$  between  $\mathbf{z}_{FN}^{E_i}$  and  $\mathbf{z}_{FN}^{E_u}$  by:

$$\widehat{\text{HSIC}}(\mathbf{z}_{FN}^{E_i}, \mathbf{z}_{FN}^{E_u}) = (n-1)^{-2} \text{tr}(K_{\mathbf{z}_{FN}^{E_i}} H K_{\mathbf{z}_{FN}^{E_u}} H)$$

where  $K_{\mathbf{z}_{FN}^{E_i}}$  and  $K_{\mathbf{z}_{FN}^{E_u}}$  are kernel matrices derived from  $\mathbf{z}_{FN}^{E_i}$  and  $\mathbf{z}_{FN}^{E_u}$ , and  $H = \mathbf{I} - \frac{1}{n} \mathbf{1}\mathbf{1}^T$  [18]. The selected seen scenario  $E_{sc}$  is formulated as:

$$E_{sc} = \arg \max_i \widehat{\text{HSIC}}(\mathbf{z}_{FN}^{E_i}, \mathbf{z}_{FN}^{E_u}), \quad 1 \leq i \leq \mathcal{N}.$$

#### D. Determination of Optimal TL Strategy

Once we have selected the most relevant scenario  $S_{sc}$  to start TL from, we move forward to ranking each layer of the selected model based on the combined weights of all the neurons of that layer. As shown in Fig. 3, we consider the neural network model of selected scenario  $S_{sc}$  has  $L_{E_{sc}}$  layers. Those layers have  $\{l_1, l_2, \dots, l_{L_{E_{sc}}}\}$  neurons, respectively. We represent the layer-wise weights of the selected neural network as:  $\{\{w_1^1, w_2^1, \dots, w_{l_1}^1\}; \{w_1^2, w_2^2, \dots, w_{l_2}^2\}; \dots; \{w_1^{L_{E_{sc}}}, w_2^{L_{E_{sc}}}, \dots, w_{l_{L_{E_{sc}}}}^{L_{E_{sc}}}\}\}$ . In order to dynamically select a different number of layers for freezing, we compute a normalized sum of weights  $\{\sum_{i=1}^{l_1} \frac{w_i^1}{l_1} = W^1, \sum_{i=1}^{l_2} \frac{w_i^2}{l_2} = W^2, \dots, \sum_{i=1}^{l_{L_{E_{sc}}}} \frac{w_i^{L_{E_{sc}}}}{l_{L_{E_{sc}}}} = W^{L_{E_{sc}}}\}$  as a normalized sum of weights from all the layers and use them to calculate a freezing threshold  $\Theta$  as:  $\Theta = \sum_{l=1}^{L_{E_{sc}}} \frac{W^l}{L_{E_{sc}}}$ .

The higher the normalized weights are for a layer from the seen model, the more pertinent that layer would be in the unseen model [19]. Thus, we rank the layers based on increasing order of normalized sum of weights. We generate a subset of layers  $\mathcal{L}_{sc}$  where we freeze the layers whose normalized sum of weights is higher than the calculated threshold:

$$\mathcal{L}_{sc} = \{l \mid W^l \geq \Theta, 1 \leq l \leq L_{E_{sc}}\}$$

In this way, the selected subset of layers to freeze  $\mathcal{L}_{sc}$  are chosen dynamically, based on the specific selected scenario  $S_{sc}$ .

## V. USED DATASETS

### A. Real-world Dataset: FLASH

The Federated Learning for Automated Selection of High-band mmWave Sectors (FLASH) dataset is a real-world multimodal dataset wherein sensory data from LiDAR, camera, and GPS participate in federated learning to speed up beam selection in mmWave V2X networks [6]. The sensor suite used in FLASH is as follows: (a) two 16-channel LiDAR, (b) a side-facing camera, and (c) GPS. Overall, FLASH contains 31923 samples ( $\sim 20$  GB processed data), covering a wide variety of real-world vehicle-to-BS LOS and NLOS scenarios.

Central to our implementation, the FLASH dataset consists of four different *categories* of data, each consisting of a LOS or NLOS connection and the potential presence of an obstacle between the Rx and BS: (a) **Cat. 1:** LOS with no obstacles, (b) **Cat. 2:** NLOS with a pedestrian, (c) **Cat. 3:** NLOS with a static car, and (d) **Cat. 4:** NLOS with a moving car. These categories are intended to comprehensively represent the environments encountered in V2X settings. More detail about these data categories, as well as the ML models they are used with, can be found in [6].

### B. Synthetic Dataset: Raymobtime

The Raymobtime dataset [10] consists of a high-fidelity virtual V2I deployment for modeling 5G and mmWave MIMO channel propagation within environments where buildings flank the two sides of a road. Using software such as Simulator for Urban MObility (SUMO), Blender, Blender Sensor Simulation (BlenSor), and Remcom Wireless Insite, a variety of moving vehicles are generated to form unique traffic patterns near a roadside Tx while image, LiDAR, and ray-tracing data are collected. Overall, 256 beam configurations, consisting of 32 codebook elements at the Rx and eight elements at the Tx, are collected, summing to over 6000 scenes ( $\sim 55$  GB data).

## VI. RESULTS

In this section, we perform experiments with TUNE on the two datasets presented in Sec. V.

1) *Implementation Details:* We utilize categorical cross-entropy loss for training with a batch size of 32 for 100 training epochs, implementing early experiment stopping if the training accuracy does not increase within 10 consecutive epochs, and use the Adam [20] optimizer with a learning rate to 0.0001. Drawing from the results of [6], [7], we assume the complexity of the categories increases in order from 1 to 2 to 4 to 3; therefore, seeing all modality data from FLASH, we perform 4, 8, 64, or 16-way classification, with the classes set to the number of available beam sectors, depending on if the experimented category is 1, 2, 3, or 4, respectively. We choose these specific numbers of sectors to highlight TL performance in environments with different numbers of beams and for ease of computation. A default 80% of the data from

one category is used in training (unless otherwise specified in each experiment), with the remaining 20% being partitioned as 10% for each of the validation and testing partitions.

2) *Performance Metrics*: TUNE evaluates each model once per epoch after training. We quantify results for each experiment by the highest test accuracy over all epochs, as well as the HSIC value between the transferred category data and experimented category data when relevant.

#### A. Competing Methods

1) **No transfer learning (S0)**: We train and test the model with all the available data without using any prior knowledge. For this strategy, all data from one category is used for training. This is used to generate a model which the ML framework can use for the other TL strategies.

2) **Load and test (S1)**: Using the model weights generated from training from other categories and strategies, we only perform testing with data from the unseen scenario. The model is not retrained or fine-tuned with the new data at any time.

3) **Load and retrain (S2)**: By first stripping the weights generated via training from previous data and categories, we retrain the entire model with the data from the unseen scenario, then perform testing with the new data.

4) **Optimal TL (S3)**: TUNE dynamically selects the most pertinent layers and freezes them for transferring the knowledge from, following the layer selection technique described in Sec. IV-D. We then retrain the rest of the layers of the model with and test the model with the data from the unseen scenario.

#### B. Learning over Different Environment with Varied Beams at the Base Station

Trans. Cat. \ Exp. Cat.	1 (Acc. (%))	2 (Acc. (%))	3 (Acc. (%))	4 (Acc. (%))
1	-	56.38	54.09	52.29
2	63.10	-	50.36	58.89
3	62.70	56.63	-	57.89
4	58.87	54.50	56.27	-

TABLE I: Testing accuracies of different categories with different numbers of possible beams using S1, transferring the model obtained by S0 from another category.

To set a comparison baseline for the other experiments, we evaluate the performance of our framework by performing S1 between all categories, with models generated by S0. As shown in Tab. I, loading or transferring the Cat. 3 model generally gives the best performance with an average accuracy of 59.07% across all categories, while the Cat. 1 model has the worst performance with an average accuracy of 54.25%. Conversely, testing on Cat. 1 data generates the highest average accuracy when experimented on at 61.56%, while Cat. 3 data gives the lowest average accuracy at 53.57%. Without retraining, these results give us a simple glimpse of the complexity of the environments each category represents and a basic expectation of how the rest of the results will proceed.

*Observation 1*: Simply loading a pre-trained model and testing the model on data from an unseen environment without retraining does not yield competitive testing accuracies.

#### C. Selection of the Knowledge to be Transferred from Previously Seen Environments

Trans. Cat.	S0 Acc. (%)	Exp. Cat.	HSIC	S1 Acc. (%)	S3 Acc. (%)
1	96.20	2	<b>6.28E-3</b>	44.13	<b>69.25</b>
		3	4.32E-4	38.82	58.17
		4	2.72E-3	37.66	64.31
2	97.49	1	<b>4.64E-3</b>	56.85	<b>71.98</b>
		3	4.13E-4	33.89	55.53
		4	2.45E-3	39.31	64.31
3	96.89	1	4.55E-3	57.46	72.78
		2	<b>6.36E-3</b>	48.38	<b>70.00</b>
		4	2.45E-3	37.99	67.59
4	96.83	1	4.90E-3	56.45	71.98
		2	<b>6.65E-3</b>	44.00	<b>67.75</b>
		3	4.16E-3	34.62	57.09

TABLE II: Accuracy and HSIC values for one transfer, transferring category models generated using S0 to experimented categories using S1 or S3. The highest accuracy experimented category results per transferred category is in bold. The S1 and S3 results are generated using 10% of the category data for training.

In this set of experiments, we analyze the best category to initialize TL from, assuming data from all categories are simultaneously available to the framework and using data from each category without redundancy. With the models generated by S0, we compare the testing accuracies and HSIC values employed via S3 and 10% of the data for training to provide some variation in accuracy. These experiments guide us by 1) helping us select a defined ‘path’ between each of the four categories for the highest efficacy TL in the FLASH dataset and 2) confirming that HSIC values are a reliable indicator of choosing the optimal pair of data categories that will increase overall TL efficacy, i.e., there exists some correlation between accuracy and HSIC values.

Immediately, we observe that testing accuracy and HSIC scores are directly correlated. When normalizing HSIC vs. accuracy, we obtain a correlation coefficient of  $r = 0.956$ . The average HSIC values for each category are 4.70E-3, 6.43E-3, 4.20E-4, and 2.54E-3 for Cat. 1, 2, 3, and 4, respectively, and transferring Cat. 3 provided the highest average S3 accuracy of 70.12% when tested with data from all other categories. With this information, we note that when performing multiple transfers, selecting the category with the *lowest* HSIC value for each transfer until the last transfer, for when we switch to selecting the highest HSIC value, will generally yield good, if not the best, results. As an example, when transferring data by using all categories one-by-one, the overall best accuracy was generated by subsequently selecting Cat. 3, 2, 4, and 1, with a final accuracy of 72.88%; in comparison, the aforementioned strategy yielded the second highest accuracy of 71.97% with a subsequent selection of Cat. 3, 4, 2, and 1. For comparison between methods, we include Tab II, generated by using S1

with 10% training data, to show the effectiveness of retraining using S3.

*Observation 2:* Given that data from multiple categories are available, experimenting on the category with the highest HSIC value will generally yield the best results.

#### D. Transferring Knowledge from the Synthetic Environment

Trans. Cat.	Exp. Cat.	S0 (Acc. (%), epochs)	TL Strategy	w/ TL (Acc. (%), epochs)
-	Synth.	29.99, 100	-	-
Synth.	1	<b>85.67</b> , 71	S1	33.34, 97
			S2	<b>86.65</b> , 57
Synth.	2	<b>86.41</b> , 61	S1	41.70, 96
			S2	<b>91.77</b> , 100
Synth.	3	<b>86.31</b> , 76	S1	36.40, 99
			S2	<b>91.14</b> , 79
Synth.	4	<b>86.23</b> , 69	S1	44.14, 100
			S2	<b>90.71</b> , 69

TABLE III: Transferring the Synthetic data model to FLASH categories. Results of interest are in bold. While transferring the knowledge from a synthetic to a real environment, general performance is preserved or improved while retraining, though convergence times may vary.

Finally, we explore the effects of the different training strategies by transferring a model generated by the data from the synthetic Raymobtime [10] dataset to each of categories in the FLASH dataset, with results shown in Tab. III. Firstly, we generate such a model using S0, and perform S0, as well as TL with S1 and S2 with this model on each of the different categories. Since the environments in the real-world and synthetic environments are drastically different, there is only a small chance that layers pertinent to the models generated from both environments exist, so we do not perform S3. In general, we observe that performing S2 yields the highest accuracy among all the methods with an average accuracy increase of 2.71% and average increase in the number of epochs elapsed by seven due to early stopping across all S2 runs when compared to S0 runs.

*Observation 3:* When transferring the model generated from a synthetic environment, TUNE yields better performance via S2 versus S0 with a comparable number of training epochs.

## VII. CONCLUSION

In this paper, we explored the impacts of using a few different TL and statistical dependency techniques with the goal of improving mmWave V2X beam selection performance. Across all TL experiments, we make three key observations: (1) simply loading a model trained on a specific V2X environment and testing said model on the data collected from a different environment without any form of retraining does not yield desirable results; (2) using a measure of statistical dependency, HSIC, and assuming that data from multiple environments is simultaneously available, selecting the environment which yields the highest HSIC value will generally yield the best accuracy as well; and (3) when transferring data from a synthetic environment and testing on data from a real-world environment, TL generally outperforms non-TL techniques by a slight margin while not requiring a lot of new data.

## ACKNOWLEDGEMENT

The authors acknowledge the funding from the US National Science Foundation (grants CNS-2120447 and CNS-2112471).

## REFERENCES

- [1] K. Li, Y. Naderi, U. Muncuk, and K. R. Chowdhury, "iSurface: Self-Powered Reconfigurable Intelligent Surfaces with Wireless Power Transfer," *IEEE Communications Magazine*, vol. 59, no. 11, pp. 109–115, 2021.
- [2] Qualcomm, "5G mmWave," <https://www.qualcomm.com/research/5g/5g-nr/mmwave>, 2022.
- [3] V. Va, J. Choi, T. Shimizu, G. Bansal, and R. W. Heath, "Inverse Multipath Fingerprinting for Millimeter Wave V2I Beam Alignment," *IEEE Trans. on Vehicular Tech.*, vol. 67, no. 5, pp. 4042–4058, 2017.
- [4] M. Alrabeiah, A. Hredzak, and A. Alkhateeb, "Millimeter Wave Base Stations with Cameras: Vision-Aided Beam and Blockage Prediction," in *IEEE Vehicular Technology Conf. (VTC2020)*, 2020, pp. 1–5.
- [5] 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.
- [6] B. Salehi, J. Gu, D. Roy, and K. Chowdhury, "FLASH: Federated Learning for Automated Selection of High-band mmWave Sectors," in *IEEE INFOCOM*, 2022.
- [7] J. Gu, B. Salehi, D. Roy, and K. R. Chowdhury, "Multimodality in mmWave MIMO Beam Selection Using Deep Learning: Datasets and Challenges," *IEEE Communications Magazine*, vol. 60, no. 11, pp. 36–41, 2022.
- [8] 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 Trans. on Vehicular Tech.*, 2022.
- [9] 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.
- [10] A. Klautau, P. Batista, N. Gonzalez-Prelcic, Y. Wang, and R. W. Heath, "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning," *2018 Information Theory and Applications Workshop (ITA)*, 2018.
- [11] M. B. Mashhadi, M. Jankowski, T.-Y. Tung, S. Kobus, and D. Gündüz, "Federated mmWave Beam Selection Utilizing LIDAR Data," *IEEE Wireless Communications Letters*, vol. 10, no. 10, pp. 2269–2273, 2021.
- [12] M. Zecchin, M. B. Mashhadi, M. Jankowski, D. Gündüz, M. Kountouris, and D. Gesbert, "LIDAR and Position-Aided mmWave Beam Selection With Non-Local CNNs and Curriculum Training," *IEEE Trans. on Vehicular Tech.*, vol. 71, no. 3, pp. 2979–2990, 2022.
- [13] 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.
- [14] S. Rezaie, A. Amiri, E. de Carvalho, and C. N. Manchón, "Deep Transfer Learning for Location-Aware Millimeter Wave Beam Selection," *IEEE Communications Letters*, vol. 25, no. 9, pp. 2963–2967, 2021.
- [15] M. Askarizadeh, M. Hussien, M. Zare, and K. K. Nguyen, "Optimized Transfer Learning For Wireless Channel Selection," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.
- [16] 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.
- [17] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Intl. conference on algorithmic learning theory*. Springer, 2005, pp. 63–77.
- [18] Z. Wang, T. Jian, A. Masoomi, S. Ioannidis, and J. Dy, "Revisiting Hilbert-Schmidt Information Bottleneck for Adversarial Robustness," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 586–597.
- [19] "Weight (Artificial Neural Network)," 2019. [Online]. Available: <https://deeppai.org/machine-learning-glossary-and-terms/weight-artificial-neural-network>
- [20] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>