

# Data-Free Knowledge Distillation for LiDAR-Aided Beam Tracking in MmWave Systems

Abolfazl Zakeri, *Member, IEEE*, Nhan Thanh Nguyen, *Member, IEEE*, Ahmed Alkhateeb, *Senior Member, IEEE*, and Markku Juntti, *Fellow, IEEE*

**Abstract**—Multimodal sensing reduces beam training overhead but is constrained by machine learning complexity and dataset demands. To address this, we propose a data-free (DF) knowledge distillation (KD) framework for efficient LiDAR-aided mmWave beam tracking, i.e., predicting the best current and future beams. Specifically, we propose a knowledge inversion framework, where a generator synthesizes LiDAR input data from random noise, guided by a loss function defined on the features and outputs of a pre-trained teacher model. The student model is then trained using the synthetic data and knowledge distilled from the teacher. The generator’s loss combines three terms, called metadata loss, activation loss, and entropy loss. For student training, in addition to the standard Kullback-Leibler divergence loss, we also consider a mean-squared error (MSE) loss between the teacher and student logits. Simulation results show that the proposed DF-KD (slightly) outperforms the teacher in Top-1 and Top-5 accuracies. Moreover, we observe that the metadata loss contributes significantly to the generator’s performance, and that the MSE loss for the student can effectively replace the standard KD loss while requiring fewer fine-tuned hyperparameters.

## I. INTRODUCTION

Beam training has long been a critical task for the performance of massive multiple-input multiple-output (mMIMO) communications. This becomes more crucial in millimeter-wave (mmWave) systems, where communication links are highly sensitive to path loss and blockages. Leveraging multimodal sensory data, such as visual, light detection and ranging (LiDAR), and radar measurements, for communication tasks, referred to as *multimodal sensing-aided communications*, has emerged as a promising approach to enhance beam training [1]–[4]. It can substantially reduce beam training overhead [5], [6] and improve beam alignment in connected vehicle environments [7]. The benefits are particularly pronounced in high-mobility scenarios, where proactive line-of-sight link prediction and future beam selection become critical for maintaining reliable connectivity.

Building on this premise, multimodal sensing-aided beamforming has attracted significant attention recently [5], [8]–[16]. For instance, Patel *et al.* [5] showed sensor-aided multimodal deep learning can efficiently estimate beamspace channels for multi-user mmWave MIMO, enabling interference-free beamforming and significantly improving spectral efficiency. Jiang *et al.* [10] demonstrated that LiDAR-aided machine learning (ML) can accurately predict and track optimal mmWave beams in real-world vehicular scenarios, noticeably reducing beam training overhead; this approach was further extended to multimodal-aided beam prediction in vehicular networks in [12], [14] and

recently to integrated sensing and communications [15].

Most existing works rely on ML to map (multimodal) sensory data to optimal beams, but such approaches are often hindered by high training costs, model complexities, limited datasets, and memory requirements. Park *et al.* [16] proposed a resource-efficient learning approach to transfer multimodal knowledge, i.e., from a highly complex model, to a monomodal, low-complexity network efficiently using a knowledge distillation (KD) framework [17]. However, this still requires substantial (multimodal) training data, which may be unavailable in practice at the UE side due to, e.g., storage limitations and at the base station (BS) side due to, e.g., UE privacy concerns when collecting raw user data. To address this, we propose a *data-free* (DF) *knowledge distillation* (KD) [18] method for beam tracking that eliminates the need for training datasets at the BS,<sup>1</sup> while still benefiting from KD to produce an efficient, low-complexity ML model [17].

We consider the LiDAR-aided beam tracking problem in [10], where the aim is to find a mapping from a sequence of previously observed LiDAR data to the optimal *current and future* beams. The setup involves a vehicle-to-infrastructure communications scenario with a single BS equipped with multiple antennas and a LiDAR sensor, and a single-antenna UE (vehicle). The goal is to develop an efficient ML model for the beam tracking, achieving high performance with low complexity, *without* access to data at the BS. To this end, we formulate the beam tracking problem as an ML task and develop a DF-KD method for it. Specifically, we propose a knowledge inversion framework, where a generator synthesizes LiDAR input data from random noise, guided by a custom loss defined on the features and outputs of a pre-trained teacher model. To train the student, we further introduce a mean-squared error (MSE) loss between the teacher and student logits in addition to the standard Kullback-Leibler (KL) divergence loss. Simulation results demonstrate the effectiveness of the proposed DF-KD, highlighting (i) the significant impact of the generator loss and (ii) that the MSE loss can replace the standard KD loss while requiring fewer hyperparameters. To the best of our knowledge, this is the first work to study DF-KD for the LiDAR-aided beam tracking problem.

## II. SYSTEM MODEL AND PROBLEM FORMULATION

### A. System Model

We consider a mmWave communications system with a BS, located at a fixed position, and a mobile UE. The BS is equipped with a uniform linear array (ULA) with  $N$  antennas, and the UE is equipped with a single-antenna receiver. Moreover, the BS is

A. Zakeri, N. T. Nguyen, and M. Juntti are with the Centre for Wireless Communications (CWC), University of Oulu, Oulu 90014, Finland, Emails: {abolfazl.zakeri; nhan.nguyen; markku.juntti}@oulu.fi. A. Alkhateeb is with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Email: alkhateeb@asu.edu. The implementation code is available at [GitHub](#).

<sup>1</sup>This is because the original validation data used in this paper resides at the BS, but our DF-KD framework is equally applicable to settings where training data collection and beam tracking occur at the UE.

also equipped with a LiDAR sensor to sense the environment and use the sensing information for communications.

At time slot  $t = 1, 2, \dots$ , the BS transmits the data signals to the UE using the beamforming vector  $\mathbf{w}(t) \in \mathbb{C}^{N \times 1}$ . We consider a codebook-based beamforming design, i.e.,  $\mathbf{w}(t) \in \mathcal{W}$ , where  $\mathcal{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$  is the codebook of all possible beamforming vectors with size  $M$ . Using analog beamsteering, the beamforming vectors in  $\mathcal{W}$  have constant modulus, i.e., each vector satisfies  $\|\mathbf{w}_m\|_2^2 = 1$ ,  $m \in \{1, \dots, M\}$ .

Denote  $\mathbf{h}(t) \in \mathbb{C}^{N \times 1}$  as the channel vector between the BS and the UE at slot  $t$ . We adopt a block-fading channel model and express the channel at slot  $t$  as

$$\mathbf{h}(t) = \sum_{l=1}^L \alpha_l(t) \mathbf{a}(\theta_l(t), \phi_l(t)), \quad (1)$$

where  $L$ ,  $\alpha_l(t)$ ,  $\theta_l(t)$ , and  $\phi_l(t)$  are, respectively, the number of paths, complex gain, azimuth angle, and elevation angle of the  $l$ -th path at time  $t$ . Furthermore, let  $s(t) \in \mathbb{C}$  be the transmit (data) signal to the UE with  $\mathbb{E}\{|s(t)|^2\} = 1$ . The received signal by the UE is then given by

$$y(t) = \mathbf{h}^H(t) \mathbf{w}(t) s(t) + n(t), \quad (2)$$

where  $n(t) \in \mathbb{C}$  is additive white Gaussian noise (AWGN) following the distribution  $\mathcal{CN}(0, \sigma^2)$ , with  $\sigma^2$  denoting the noise variance at the UE's receiver.

Given a predefined codebook, the beamforming vector  $\mathbf{w}(t) \in \mathcal{W}$  is specified by the beam index  $m(t) \in \{1, \dots, M\}$  at time  $t$ . The beamforming problem is then formulated as finding the optimal beam index  $m^*(t)$  that maximizes the received SNR of the UE, i.e.,

$$m^*(t) = \arg \max_{m(t) \in \{1, \dots, M\}} |\mathbf{h}^H(t) \mathbf{w}_{m(t)}|^2. \quad (3)$$

We remark that if the channel vector  $\mathbf{h}(t)$  is available, the above problem can be easily solved. In fact, the optimal beamforming is the matched filter, i.e.,  $\mathbf{w}(t) = \sqrt{P} \frac{\mathbf{h}(t)}{\|\mathbf{h}(t)\|}$ , with  $P$  denoting the transmit power budget, which is commonly known as a maximum ratio transmission beamformer. However, the main challenge here is that obtaining the channel information at all times requires excessive signaling overhead, additional latency, and resource consumption. To overcome this, similarly to, e.g., [3], [4], [14], we propose the idea of using LiDAR sensory data with ML for beamforming design. The detailed learning task is elaborated next.

**Remark 1.** We consider LiDAR because (1) it provides superior spatial 3D resolution and (2) its emergence and adaptation in autonomous vehicles. Nevertheless, LiDAR incurs substantial data collection and storage overhead, making large-scale training datasets difficult and costly to maintain, particularly at the UE, or at the BS, thereby further motivating our DF-KD approach for LiDAR-aided beam tracking.

#### B. ML Beamforming Problem Definition

The goal is to use LiDAR data to perform *beam tracking*, i.e., predict the current and future best beams. To formulate this, let  $\mathbf{x}(t) \in \mathbb{R}^D$  denote the (preprocessed) LiDAR data, with fixed-length feature of size  $D$ , captured at the BS at time  $t$ . The sequence of  $L \in \mathbb{N}$  most recent LiDAR measurement data is denoted as  $\mathcal{X}(t; L) = \{\mathbf{x}(t-L+1), \dots, \mathbf{x}(t)\}$ . Considering the

time-dependent nature of UE mobility we utilize past LiDAR data rather than relying solely on current observations from slot  $t$ . Therefore, the goal is to learn a mapping function that takes the LiDAR input sequence,  $\mathcal{X}(t; L)$ , and predicts the optimal current/future beam indices using ML.

**Beam Tracking Problem:** Let  $f_{\Theta}(\cdot)$  be a mapping function, parameterized by an ML model (i.e., the neural network),  $\Theta$ , that maps the input LiDAR data to a beam index for the current time  $t$  and  $V \in \mathbb{N}$  number of future beams  $t+1, \dots, t+V$  according to

$$\hat{m}(t+v) = \arg \max_{m \in \{1, \dots, M\}} f_{\Theta}(\mathcal{X}(t; L); v)_m, \quad (4)$$

where  $f_{\Theta}(\mathcal{X}(t; L); v) \in \mathbb{R}^M$ ,  $v \in \{0, 1, \dots, V\}$  is the output logits for each beam  $m$  of time  $t+v$ , and  $\hat{m}(t')$  is the predicted beam index at time  $t' = t, t+1, \dots, t+V$ , by the model.<sup>2</sup> The goal is to train the model to learn optimal  $f_{\Theta}^*(\cdot)$  corresponding to  $\Theta^*$  such that  $\hat{m}(t') = m^*(t')$ .

The final beam selection for each time  $t$  is generally based on top- $K$  promising beams obtained as above from the ML model. Given the top- $K$  beams, the BS can either 1) directly choose the top-1 beam and mitigate the beam training overhead, or 2) choose the  $K$  beams and perform an over-the-air beam training only for those  $K$  beams [10] with significantly reduced overhead compared to conducting an exhaustive search over the entire codebook. Next, we briefly introduce DF-KD and describe how we adapt it for the beam training task.

### III. DATA-FREE KD FOR BEAM TRACKING

#### A. Main Ideas

DF-KD builds upon the foundation of conventional KD, but removes the requirement of having access to the training dataset [18]. In standard KD [17], the objective is to transfer the knowledge from a large, high-capacity model, called *teacher* and denoted by  $T$ , to a more compact model, called *student* and denoted by  $S$ , typically one with fewer parameters or a simpler architecture, while retaining as much performance as possible. This is often achieved by minimizing the divergence between the student's output distribution and the "soft" output probabilities of the teacher, which are computed using a temperature parameter,  $T$ , to soften the logits. Beyond output probabilities, KD can also transfer knowledge at different representational levels, such as intermediate hidden features or relational structures between samples [19].

In practice, the training samples may be unavailable due to constraints on privacy, storage, or transmission overheads. Instead, *synthetic data* or feature priors, often guided by the *pretrained* teacher's internal activation statistics, are generated and used for distillation. This enables knowledge transfer without direct access to the training datasets. Motivated by this, we propose a DF-KD framework that leverages a pretrained teacher model and feature statistics—without requiring the original training dataset (i.e., LiDAR inputs or ground-truth labels)—to first train a synthetic data generator and then train a student model. The procedure is summarized in Alg. 1, which

<sup>2</sup>This is the same as using the maximum likelihood where the probability distribution over the beam class is obtained by applying softmax to the logits.

---

**Algorithm 1:** Data-Free KD Algorithm
 

---

**Initialize :** Pretrained teacher model  $T$  and metadata, generator model  $G$ , student model  $S$

1 /\* Step (1): Train generator via knowledge inversion \*/

2 **for** each epoch **do**

3   Sample noise vector for each batch:  $\tilde{x} \sim \mathcal{N}(0, I)$

4   Generate synthetic samples:  $\hat{x} \leftarrow G(\tilde{x})$

5   Compute teacher logits and features from  $T(\hat{x})$

6   Compute the loss function  $\mathcal{L}^G$

7   Update generator parameters (i.e., backpropagation)

8 /\* Step (2): Train the student using synthetic data \*/

9 **for** each epoch **do**

10   Sample noise vectors for each batch:  $y \sim \mathcal{N}(0, I)$

11   Generate synthetic samples:  $\hat{x} \leftarrow G(y)$

12   Get teacher logits:  $z_T \leftarrow T(\hat{x})$

13   Get student logits:  $z_S \leftarrow S(\hat{x})$

14   Compute the loss either by (9) or (10)

15   Update student parameters (i.e., backpropagation)

**Output :** Trained student

---

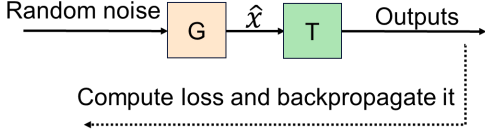


Fig. 1: A schematic of training the generator  $G$  using the pretrained teacher  $T$

takes the pretrained teacher and statistical priors as input and proceeds in two main steps. In step (1), the generator  $G$  is trained using the pretrained teacher. In step (2), the student model is trained using both the pretrained teacher and the generator. The details of these two steps are explained in the following sections.

### B. Synthetic Data Generation

To construct meaningful LiDAR-type input data, denoted by  $\mathcal{X}$ , for training the student, we use a knowledge inversion framework where a data generator model  $G$  is trained from random noise and a loss function defined on the feature and final outputs (logits) of the pretrained teacher. A schematic of this procedure is depicted in Fig. 1. For the generator, we use a (simple) two-layer fully connected feedforward neural network with ReLU and tanh activations, respectively, in the first and the output layer. The model takes in a random noise vector and generates synthetic LiDAR-like sequences. The output is shaped as a sequence of  $K$  frames, each with  $D$  features matching to the real LiDAR input format. Then, a zero-padding is applied with length  $V$ , the same as in the real dataset [10], to properly account for the  $V$  future slot beam prediction. The generated synthetic inputs are then passed through the pretrained teacher model to obtain output logits and intermediate features for loss computation. The loss function can be designed based on the intuition that high-quality synthetic data, i.e., data that closely resembles real samples, should induce low-entropy predictions and strong alignment between the current teacher features and those obtained during training on the real dataset.

**Generator Loss Function:** We introduce several different loss functions and examine their significance in the numerical results Section IV.

**Metadata Loss:** The first loss, which we refer to as the metadata loss, captures the feature mismatch between the teacher model's activations when processing synthetic data versus real LiDAR data. This means that along the model, metadata is also saved [18]. The metadata we use here is the statistics of the features of the teacher model, i.e., the mean and variance of the hidden state of the last layer of the teacher model, respectively denoted by  $\mu$  and  $\sigma^2$ . Then we define the metadata loss as

$$\mathcal{L}_{\text{meta.}}^G \triangleq \text{MSE}(\mu, \hat{\mu}) + \text{MSE}(\sigma^2, \hat{\sigma}^2), \quad (5)$$

where  $\hat{\mu}$  and  $\hat{\sigma}^2$  are the mean and variance of the teacher's features passed through the synthetic data.

**Activation Loss:** We define an activation regularization loss that maximizes the activation norm, encouraging the generator to produce inputs that elicit strong and meaningful responses from the teacher model. Strong activations typically correlate with in-distribution behavior [20], thereby guiding the generator toward creating informative synthetic samples that facilitate effective student learning in our DF-KD framework. Formally, it is defined as

$$\mathcal{L}_{\text{act.}}^G = -\frac{1}{B} \sum_{i=1}^B \left\| \tilde{f}_T(G(\tilde{\mathbf{X}})) \right\|_2, \quad (6)$$

where  $G(\cdot)$  is the synthetic input data generated from a latent  $\tilde{\mathbf{X}} \in \mathbb{R}^{B \times L}$  with  $\tilde{X}_{ij} \sim \mathcal{N}(0, 1)$ ,  $\tilde{f}_T(\cdot)$  denotes intermediate feature activations from the teacher network,  $B$  is the batch size, and  $L$  is the generator input size.

**Entropy Loss:** Lastly, we consider the entropy loss from the probability distribution obtained using softmax applied to the teacher's logits, which is defined as:

$$\mathcal{L}_{\text{ent.}}^G = -\sum_{i=1}^M p_i \log p_i \quad (7)$$

where  $p_i = \text{softmax}(z_i) \triangleq \frac{e^{z_i}}{\sum_{j=1}^M e^{z_j}}$  is the predicted probability for beam  $i$  and  $z_i$  is the logit (i.e., raw output of the model) for beam  $i$ . Entropy loss measures the uncertainty or confidence of the model's predictions. A high entropy value indicates that the model is uncertain (i.e., its predictions are spread out across classes), while a low entropy implies confident predictions (i.e., one class has a dominant probability).

Eventually, the final loss for training the generator is given by

$$\mathcal{L}^G = \mathcal{L}_{\text{meta.}}^G + \alpha \mathcal{L}_{\text{act.}}^G + \beta \mathcal{L}_{\text{ent.}}^G, \quad (8)$$

where  $\alpha$  and  $\beta$  are positive hyperparameter, emphasizing the corresponding loss.

### C. Training the Student

Once the generator is trained, it is used to produce synthetic data for performing KD using the standard procedures, as schematically depicted in Fig. 2. However, since no ground-truth labels are available for the synthetic data, the cross-entropy loss, as in standard KD [17], cannot be applied. Instead, the Kullback-Leibler (KL) divergence loss between the teacher's and student's output distributions remains a valid and effective



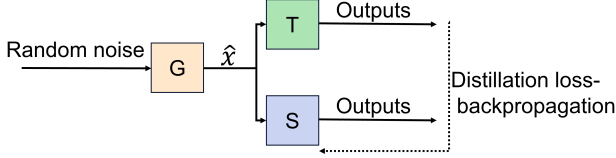


Fig. 2: A schematic of training the student  $S$  using the pretrained teacher  $T$  and pretrained generator  $G$

choice. Additionally, inspired by [21], we also incorporate a mean squared error (MSE) loss between the teacher's and student's logits. We numerically compare the performance of the KL and the MSE losses in Section IV.

**KL Divergence Loss:** KL divergence measures how one probability distribution diverges from another. In KD, it aligns the student's output probability distribution with the teacher's. Given the teacher and student logits softened by a temperature parameter  $T$ , the KL divergence between the resulting softmax distributions is computed as

$$\mathcal{L}_{\text{KL}} = \frac{T^2}{B} \sum_{j=1}^B \sum_{i=1}^M p_{ij} \log \left( \frac{p_{ij}}{q_{ij}} \right) \quad (9)$$

where  $p_{i,j} = \text{softmax}(z_{i,j}^T/T)$  is the teacher's soft probability for beam  $i$  in batch  $j$  and  $q_{i,j} = \text{softmax}(z_{i,j}^S/T)$  is the student's soft probability for beam  $i$  in batch  $j$ . Notice that a higher temperature  $T$  softens the distributions, allowing the student to better capture the relative class similarities learned by the teacher.

**MSE Loss Between Logits:** We further introduce an MSE loss, in training the student, that directly matches the raw logits (pre-softmax outputs) of the teacher and the student, defined as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{B \cdot M} \sum_{j=1}^B \sum_{i=1}^M (z_{j,i}^S - z_{j,i}^T)^2. \quad (10)$$

This loss encourages the student to replicate the structure and scale of the teacher's output distribution without relying on label information. Furthermore, unlike the KL loss, it does not necessitate tuning of the temperature parameter  $T$ .

#### IV. NUMERICAL RESULTS

This section provides simulation results to compare the performance of the proposed DF-KD Alg. 1 as well as to examine the impact of different losses, defined for the generator and the student training, on the overall performance. Below, we first explain the simulation scenario and parameters.

**Scenario:** Similarly to [10], we choose a vehicular-to-infrastructure communication Scenario 8 from the DeepSense dataset [1], where the vehicle speed in the dataset ranges from 3 to 18 km/h. The moving vehicle carries an omnidirectional mmWave 60 GHz transmitter to communicate with the BS. The BS is equipped with a 60 GHz receiver that has a 16-element phased array and uses a pre-defined beam codebook of 64 beams, i.e.,  $M = 64$ . Further details about the dataset and the scenario can be found in [1], [10].

**Model and Parameters:** We adopt the model proposed in [10] as the teacher, which is a best-performing model with an overall loss of 1.3172 in the test. The observation window is  $L = 8$ , and the future prediction beam length  $V = 3$ . Therefore, the model output predicts the current and 3 future slots' best beam. We reduce the number of hidden neurons of the teacher

from 128 in [10] to 32 for the student model; this results in 96,640 trainable parameters of the teachers reduced to 25,408 for the student. For the generator model, we set the number of input neurons (i.e., length of the random data input) to 500 and the number of hidden-layer neurons to 64. For all the models, the batch size is 32 and the optimizer is Adam with a learning rate of  $10^{-3}$ . For DF-KD, both the generator and the student are trained for 500 epochs. For the performance criteria, Top-1 and Top-5 accuracy are shown.

**Impact of the generator loss:** In Fig. 3, we analyze the effect of different generator training losses introduced in Sec. III-B, while keeping the student training loss fixed as the KL loss (9) with a fine-tuned temperature of  $T = 5$ . We evaluate four variants: (i) weighted loss in (8), (ii) metadata loss only, (iii) activation loss only, and (iv) entropy loss only, to assess their individual contributions. The results clearly highlight the critical role of effective generator training, as the quality of the synthesized (fake LiDAR) data directly influences the student's learning performance. Among the standalone losses, the metadata loss yields substantially better results compared to the activation and entropy losses. This can be attributed to the fact that the metadata loss leverages statistical information extracted from the teacher model when training the generator, thus providing more informative guidance for producing realistic and relevant synthetic data.

**Impact of the student loss:** Fig. 4 examines the effect of the KL loss (9) and the proposed MSE loss (10) on the Top-1 and Top-5 accuracy for current and three future beam predictions of the provided DF-KD, using the best-performing generator. The results show that both KL and MSE losses enable the student to match the teacher's performance, with the MSE loss even surpassing KL in future beam predictions 2 and 3. This suggests that the MSE loss can serve as a simpler alternative to the KL loss, eliminating the need for temperature fine-tuning.

**Standard KD vs. DF-KD performance:** Fig. 5 presents the Top-1 (a) and Top-5 (b) accuracies of the teacher model and various (fine-tuned) KD methods. We also consider the student "without KD" to highlight the benefits of knowledge distillation. In our proposed DF-KD, the generator is trained with the metadata loss, and the student is trained with the proposed MSE loss. For comparison, the "standard KD" employs the KD loss from [17]:

$$\mathcal{L}_{\text{KD}} = \gamma \mathcal{L}_{\text{KL}} + (1 - \gamma) \mathcal{L}_{\text{cross-ent.}}, \quad (11)$$

where  $\mathcal{L}_{\text{KL}}$  is the KL loss in (9) and  $\mathcal{L}_{\text{cross-ent.}} = -\sum_{i=1}^M y_{\text{true},i} \log q_i$  is the cross-entropy loss, where  $y_{\text{true},i}$  is the true label for class  $i$ , such that it is 1 for the correct class and 0 otherwise; moreover,  $q_i$  is the student's soft probability for beam  $i$ . The variant "KD-MSE" refers to the standard KD but with the KL loss replaced by our proposed MSE loss.

The results reveal that the MSE loss yields comparable, and in some cases superior, beam prediction accuracies for both the standard KD and DF-KD students. However, we also observe that standard KD with MSE requires more training epochs to converge than when using the original KD loss. In Fig. 5, the standard KD with the loss in (11) converges in 20 epochs, whereas its MSE-based counterpart requires 40 epochs. Fig. 5(a) shows that DF-KD slightly outperforms its teacher (by

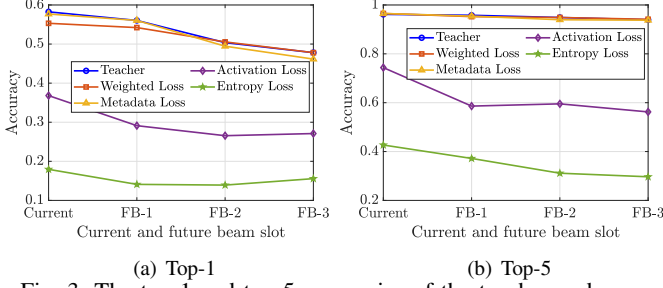


Fig. 3: The top-1 and top-5 accuracies of the teacher and proposed DF-KD with different generator losses, where the weighted loss is given by (8) with  $\alpha = 10^{-4}$  and  $\beta = 10^{-2}$

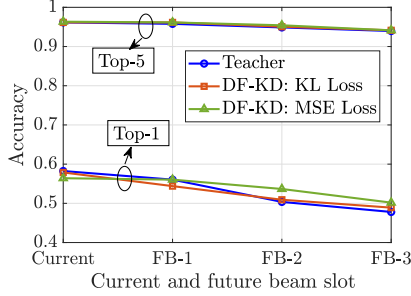


Fig. 4: The top-1 and top-5 accuracies of the teacher and proposed DF-KD for different student training loss, where the best performing generator loss is chosen from Fig. 3, where  $T = 5$

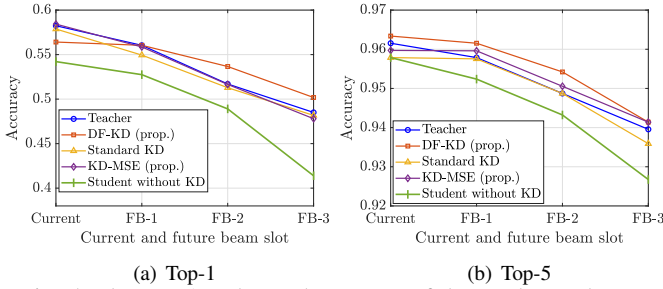


Fig. 5: The Top-1 and Top-5 accuracy of the teacher and proposed DF-KD with the MSE loss vs the standard KD and KD with MSE loss (proposed), where for the KD loss  $\gamma = 0.7$  and  $T = 5$

about 3.8%), suggesting that distillation serves as an implicit regularizer and enhances generalization [22]. Finally, Fig. 5 and Fig. 4 suggest that the KL loss can be replaced by the simpler MSE loss, with fewer hyperparameters, while maintaining, or even improving, accuracy, consistent with observations in [21].

## V. CONCLUSIONS

We developed a DF-KD method for LiDAR-aided beam tracking in vehicular networks. We considered a knowledge inversion framework to train the data generator and introduced a loss function that combines three different components. To train the student model, we further proposed an MSE loss in addition to adopting the KL divergence loss. Our results demonstrated the effectiveness of the proposed DF-KD approach and revealed two important observations: (i) the metadata loss of the generator has a significantly greater contribution to performance, and (ii) the MSE loss of the student training can replace the standard KD loss while retaining or potentially improving performance, with advantages of simplicity and minimal fine-tuning parameters. Future work will explore

multimodal datasets and more complex systems to assess the generalizability of DF-KD in beam training.

## REFERENCES

- [1] A. Alkhateeb, G. Charan, T. Osman, A. Hredzak, J. Morais, U. Demirhan, and N. Srinivas, "Deepsense 6G: A large-scale real-world multi-modal sensing and communication dataset," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 122–128, Sep. 2023.
- [2] J. Gu, B. Salehi, D. Roy, and K. R. Chowdhury, "Multimodality in mmWave MIMO beam selection using deep learning: Datasets and challenges," *IEEE Commun. Mag.*, vol. 60, no. 11, pp. 36–41, 2022.
- [3] A. Ali, N. Gonzalez-Prelcic, R. W. Heath, and A. Ghosh, "Leveraging sensing at the infrastructure for mmWave communication," *IEEE Commun. Mag.*, vol. 58, no. 7, pp. 84–89, Jul. 2020.
- [4] A. Oliveira, D. Suzuki, S. Bastos, I. Correa, and A. Klautau, "Machine learning-based mmwave MIMO beam tracking in V2I scenarios: Algorithms and datasets," in *Proc. IEEE Latin-American Conf. on Commun. (LATINCOM)*, pp. 1–5, Medellin, Colombia, Dec. 2024.
- [5] K. Patel and R. W. Heath, "Harnessing multimodal sensing for multi-user beamforming in mmWave systems," *IEEE Trans. Wireless Commun.*, vol. 23, no. 12, pp. 18725–18739, Dec. 2024.
- [6] S. Imran, G. Charan, and A. Alkhateeb, "Environment semantic communication: Enabling distributed sensing aided networks," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 7767–7786, Dec. 2024.
- [7] M. B. Mollah, H. Wang, M. A. Karim, and H. Fang, "Multi-modality sensing in mmWave beamforming for connected vehicles using deep learning," *IEEE Trans. on Cogn. Commun. Netw.*, Early Access, 2025.
- [8] A. Zakeri, N. T. Nguyen, A. Alkhateeb, and M. Juntti, "Constrained multimodal sensing-aided communications: A dynamic beamforming design," in *Proc. IEEE Global Commun. Conf.*, Accepted, 2025. Available at: <https://arxiv.org/abs/2505.10015>.
- [9] C. Zheng, J. He, C. G. Kang, G. Cai, Z. Yu, and M. Debbah, "M2BeamLLM: Multimodal sensing-empowered mmWave beam prediction with large language models," *arXiv preprint arXiv:2506.14532*, Jun. 2025.
- [10] S. Jiang, G. Charan, and A. Alkhateeb, "LiDAR aided future beam prediction in real-world millimeter Wave V2I communications," *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 212–216, Feb. 2023.
- [11] B. Salehi, G. Reus-Muns, D. Roy, Z. Wang, T. Jian, J. Dy, S. Ioannidis, and K. Chowdhury, "Deep learning on multimodal sensor data at the wireless edge for vehicular network," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7639–7655, Jul. 2022.
- [12] B. Shi, M. Li, M.-M. Zhao, M. Lei, and L. Li, "Multimodal deep learning empowered millimeter-Wave beam prediction," in *Proc. IEEE Veh. Technol. Conf.*, pp. 1–6, Singapore, Jun. 2024.
- [13] B. Salehihi Kouei, *Leveraging Deep Learning on Multimodal Sensor Data for Wireless Communication: From mmWave Beamforming to Digital Twins*. PhD thesis, Northeastern University, 2024.
- [14] G. Charan, T. Osman, A. Hredzak, N. Thawdar, and A. Alkhateeb, "Vision-position multi-modal beam prediction using real millimeter wave datasets," in *Proc. IEEE Wireless Commun. and Networking Conf.*, pp. 2727–2731, Austin, TX, USA, Apr. 2022.
- [15] K. Zhang, W. Yu, H. He, S. Song, J. Zhang, and K. B. Letaief, "Multimodal deep learning-empowered beam prediction in future THz ISAC systems," *arXiv preprint arXiv:2505.02381*, May 2025.
- [16] Y. M. Park, Y. K. Tun, W. Saad, and C. S. Hong, "Resource-efficient beam prediction in mmWave communications with multimodal realistic simulation framework," *arXiv preprint arXiv:2504.05187*, Apr. 2025.
- [17] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [18] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," in *Proc. NeurIPS Workshop Deep Learn. Without Labels*, (Long Beach, CA, USA), 2017. [Online]. Available: <https://arxiv.org/abs/1710.07535>.
- [19] A. M. Mansourian, R. Ahmadi, M. Ghafouri, A. M. Babaei, E. B. Golezani, Z. Y. Ghamchi, V. Ramezani, A. Taherian, K. Dinashi, A. Miri, and S. Kasaei, "A comprehensive survey on knowledge distillation," *arXiv preprint arXiv:2503.12067*, Mar. 2025.
- [20] Y. Wang, L. Cheng, M. Duan, Y. Wang, Z. Feng, and S. Kong, "Improving knowledge distillation via regularizing feature direction and norm," in *Lecture Notes in Computer Science*, pp. 20–37, Springer Nature, 2024.
- [21] T. Kim, J. Oh, N. Kim, S. Cho, and S. Yun, "Understanding knowledge distillation," *OpenReview (ICLR Submission)*, 2019.
- [22] Y. Shen, L. Xu, Y. Yang, Y. Li, and Y. Guo, "Self-distillation from the last mini-batch for consistency regularization," in *CVPR*, pp. 11943–11952, IEEE/CVF, Jun. 2022.