Deep Transfer Learning for Location-aware Millimeter Wave Beam Selection

Rezaie, Sajad; Amiri, Abolfazl; De Carvalho, Elisabeth; Manchon, Carles Navarro

Published in:
I E E E Communications Letters

DOI (link to publication from Publisher):
10.1109/LCOMM.2021.3091120

Publication date:
2021

Document Version
Accepted author manuscript, peer reviewed version

Deep Transfer Learning for Location-aware Millimeter Wave Beam Selection

Sajad Rezaie, Abolfazl Amiri, Elisabeth de Carvalho, and Carles Navarro Manchón

***Abstract*—The main bottleneck for using deep neural networks in location-aided millimeter wave beam alignment procedures is the need for large datasets to tune their large set of trainable parameters. This paper proposes to use the transfer learning technique in order to reduce the dataset size requirements in deep-learning based beam selection. Information transfer can be done from one environment to another, or from one antenna configuration to another, which we refer to as domain and task adaptation, respectively. Numerical evaluations show a significant gain in using transfer learning in both domain and task adaptation scenarios, especially with limited datasets.**

***Index Terms*—millimeter wave, beam alignment, deep learning, transfer learning, domain adaptation, task adaptation**

I. Introduction

While the use of highly directional transmission in mmWave communications allows for high antenna gains, it introduces the challenge of aligning beams with the direction of the line of sight (LOS) or strong non LOS (NLOS) paths. To simplify the beam alignment procedure and also the radio frequency (RF) implementation, codebook-based analog beamforming has been proposed [1]. A simple approach is to exhaustively search over all combinations of the precoder and combiner configurations; however, this procedure suffers from high overhead and latency. As an alternative to the exhaustive search, hierarchical search methods have been proposed to reduce the latency by searching directions with different beam widths [1]. They suffer, however, from a degraded accuracy when dealing with low signal-to-noise ratio (SNR) at large beam widths. Context information-based (CI-based) beam alignment methods have been proposed to exploit contextual information such as user position and speed to address these challenges. CI can be obtained by either sensors on transceivers or out-band measurements which may cause additional overheads [2]. Making use of the context information of the user and environment, probabilistic data-driven methods propose a short list of candidate beams by leveraging the information in the training data [3].

Furthermore, machine learning techniques, especially reinforcement learning (RL) and neural networks (NNs), have been used to provide more accurate beam alignment using their high capability in non-linear problems [4], [5]. Although RL is a well-behaved solution for sequential decision-making problems, it needs constant interaction with the environment to find the optimal policy [6]. NNs can predict LOS blockage and yield accurate beam alignment, significantly outperforming look-up table type of algorithms such as inverse fingerprinting [7]. NN-based approaches, however, need a large training dataset to optimize all the trainable parameters in their networks. The training datasets are specific to each deployment site and its propagation environment, as well as to the antenna array configurations used by BS and user [7], [8]. Due to the time and cost associated with measurement collection, the construction of these large training datasets constitutes the main obstacle to the use of neural networks in practice. As a solution to this problem, the transfer learning (TL) technique can be used to transfer parts of information in a network previously trained with a large dataset to another network for which only a small training dataset is available. The information can be transferred from an environment to another environment or from an antenna configuration to a different one, which are known in the research community as domain and task adaptation, respectively [9], [10].

In this paper, we propose a TL framework for location-based beam selection that optimizes the neural networks' parameters using small datasets. The proposed procedure can transfer useful information from a network trained with large datasets. We show the benefits of using TL techniques when it is not feasible to measure many samples at each environment and for each possible configuration. To benchmark the proposed method, we considered the two hierarchical beam-search (HBS) methods proposed by [1] and the deep learning (DL)-based beam alignment without TL in [7] as baselines. The numerical results, which are obtained based on ray-traced channel responses, show that the proposed method can reduce the performance gap of training with large and small datasets, and it also outperforms the baselines. Significant performance improvement is observed when applying TL in both cases: from an environment to another environment and from an antenna configuration to a different one.

II. System Model

We consider a 3-dimensional (3D) indoor downlink scenario consisting of a transmitter (TX) and a receiver (RX) equipped with uniform linear arrays (ULA) and operating in the mmWave frequency band. We assume that TX and RX ULAs are placed horizontally and are made respectively of N_t and N_r elements, with the elements separated by half a wavelength.

A. Channel Model

Without loss of generality, the transmitter is located at the origin of the coordinate system, $\boldsymbol{p}_t = (0, 0, 0)$, with fixed angle

This work is supported by the Danish Council for Independent Research, grant no. DFF 8022-00371B.

All the authors are with the Department of Electronic Systems, Aalborg University, Denmark, e-mail: {sre, aba, edc, cnm}@es.aau.dk

α_t, relative to the x-axis. The receiver is placed randomly inside the defined area at position $\boldsymbol{p}_r = (x_r, y_r, z_r)$ with a random orientation $\alpha_r \in [0, 2\pi)$. We only consider changes of the RX's ULA orientation in the azimuth plane.

The channel matrix between the transmitter and the receiver with one LOS and L non-line-of-sight (NLOS) paths, is modeled as

$$\boldsymbol{H} = \sum_{l=0}^{L} \sqrt{\rho_l}\, e^{j\vartheta_l}\, \boldsymbol{a}_r(\phi_{r,l}, \theta_{r,l})\, \boldsymbol{a}_t^H(\phi_{t,l}, \theta_{t,l}), \tag{1}$$

where $\boldsymbol{H} \in \mathbb{C}^{N_r \times N_t}$, ρ_l and ϑ_l are the receive power and the phase of the l-th path, respectively. Furthermore, $\phi_{r,l}$ and $\theta_{r,l}$ denote the azimuth and elevation angle of the AoA with respect to the receiver array axis. The antenna array response of the receiver, $\boldsymbol{a}_r$, is

$$\boldsymbol{a}_r(\phi_{r,l}, \theta_{r,l}) = \frac{1}{\sqrt{N_r}}[1, e^{j\pi sin(\theta_{r,l})\, cos(\phi_{r,l})}, \ldots \\ , e^{j\pi(N_r-1)\, sin(\theta_{r,l})\, cos(\phi_{r,l})}]^T.$$

The antenna array response of the transmitter, $\boldsymbol{a}_t$, is analogously defined using the azimuth and elevation AoDs, $\phi_{t,l}$ and $\theta_{t,l}$, which are measured relative to the TX array axis.

B. Beam Codebook

Analog phase shifters have the ability to form beams by applying phase shifts to the signal of each antenna element. For simplicity, we employ discrete Fourier transform (DFT)-based codebooks with predefined precoders and combiners that steer the arrays in the azimuth direction. We define the precoders and combiners as

$$\boldsymbol{u}_p = \boldsymbol{a}_t(\phi_p, \pi/2), \qquad p \in \{1, \ldots, N_t\},$$
$$\boldsymbol{v}_q = \boldsymbol{a}_r^H(\phi_q, \pi/2), \qquad q \in \{1, \ldots, N_r\}.$$

where ϕ_p and $\phi_q \in [0, \pi)$ are $\arccos((2p-1-N_t)/N_t)$, and $\arccos((2q-1-N_r)/N_r)$, respectively.

The sets $\mathcal{U} = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{N_t}\}$ and $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{N_r}\}$ denote the codebooks of all possible analog beamformers at the transmitter and the receiver, respectively. By applying the precoder $\boldsymbol{u}_p$ and combiner $\boldsymbol{v}_q$, the received signal strength (RSS) at the receiver may be expressed as

$$\mathbf{R}[p, q] = \left\| \sqrt{P_t} \boldsymbol{v}_q^H \boldsymbol{H} \boldsymbol{u}_p s + \boldsymbol{v}_q^H \boldsymbol{n} \right\|^2 \tag{2}$$

where P_t and $s \in \mathbb{C}$ are the transmission power and the known, unit power training symbol, respectively. Also, $\boldsymbol{n} \in \mathbb{C}^{N_r}$ denotes zero mean complex Gaussian noise variance σ_n^2.

III. Transfer Learning for Beam Selection

Exploiting propagation environment information together with the position and orientation information helps to make the beam alignment procedure more robust against blockage, along with reducing the latency of the procedure. This prior knowledge is extractable from measurements in the same environment in a training phase. However, the mapping from context information of the receiver, accounting for the geometry of the environment, to the best beam alignment is a highly non-linear function. Due to the high capacity of neural networks to learn non-linear functions, deep neural networks (DNNs) are suitable for our problem. While DNN needs a large dataset to tune its parameters, capturing many samples at each environment is not feasible. TL is a technique that transfers the knowledge learned in a situation, usually with a large dataset, to improve the performance of learner in a new situation, usually with a smaller dataset.

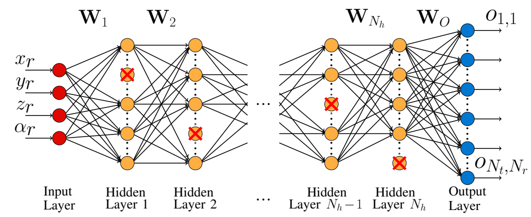


Fig. 1. Deep neural network architecture of the beam alignment method using receiver coordinates and orientation as context information.

A. Network Architecture

Fig. 1 shows a proposal structure of a feedforward, fully-connected, deep neural network that predicts probabilities of each beam pair resulting in the largest RSS according to the coordinates and orientation of the receiver. This network design is inspired by the structure in [7], where each output corresponds to a unique beam pair. Thus, there are $N_t N_r$ outputs in the proposed network. Also, $tanh$ and $softmax$ functions, respectively, are used as the non-linear activation functions of the hidden and output layers. The outputs of the deep neural network, $\mathbf{O}$, are the nonlinear functions of the input vector $\mathbf{x}$ as

$$\mathbf{O} = f^{(O)}_{\mathbf{W}_O} \circ f^{(N_h)}_{\mathbf{W}_{N_h}} \circ \ldots f^{(1)}_{\mathbf{W}_1}(\mathbf{x}) \tag{3}$$

where $h \circ g(x) = h(g(x))$. $f^{(i)}_{\mathbf{W}_i}$, $i = 1, \cdots, N_h$ and $f^{(O)}_{\mathbf{W}_O}$ denote the nonlinear transformation function of the N_h hidden layers and output layer, respectively. In addition, $\mathbf{W}_i$, $i = 1, \cdots, N_h$ denotes the matrix of all the trainable weights at the i-th hidden layer, and $\mathbf{W}_O$ includes all the trainable weights at the output layer of the network. Consider the set $\mathcal{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_{N_h}, \mathbf{W}_O\}$ which includes all the trainable parameters of the network. The number of trainable parameters depends on the number of neurons in each layer and also the number of considered hidden layers in the structure. According to the previous research, there are hundreds of thousands trainable weights in a fully connected neural network with enough capacity for mapping context information to beam pairs [4], [5], [7]. However, training a large number of parameters requires a large dataset, between $10,000$ to $100,000$ samples from the environment with a given antenna configuration at the transceivers [4], [7], which is a time- and cost-consuming operation.

To generate the training dataset of environment Ξ with M^Ξ user points, we calculate each beam pair's RSS using (2) for each user position and orientation in the user grid. The dataset $\mathbb{D}^\Xi = \{(\mathbf{x}^m, L^m)\}, m = 1, \cdots, M^\Xi$, is composed of M^Ξ pairs, where $\mathbf{x}^m$ and L^m, respectively, are the vector of the NN's input ($[x_r, y_r, z_r, \alpha_r]$) and the index of beam pair with the highest RSSs for m-th user. As L^m is the post-processed version of the received signals for all beam pairs, it implicitly contains the information about geometry and size of the environment. As an example, when the LOS is blocked, other strong paths determine the beam pair with the highest RSSs. So, the NN can learn, for a given location and orientation, which beam pairs have higher probability to be the best choice for transmission [7].

B. Transfer Learning

Transfer learning can be applied across domains and tasks, where the goal is to transfer knowledge between different environments or different tasks, respectively [9].

1) Domain Adaptation: Assume we have a large dataset of measurements in an environment, which we refer to as source domain, with N_t and N_r antenna elements at the TX and RX, respectively. Also, consider a network NET^S with ${N_h}^S$ and n^S hidden layers and neurons at each hidden layer, respectively, predicting the best beam pair accurately. Due to the availability of a large dataset, $\mathbb{D}^S$, the weights of the network NET^S can be initialized with random values and be trained by a standard backpropagation algorithm. On the other hand in a second environment, which we refer to as destination domain, only a limited-size dataset, $\mathbb{D}^D$, with the same antenna configuration as the source domain is available. If we consider another network NET^D with the same number of layers and neurons as NET^S, there is a chance to reuse $\mathcal{W}^S$, the trained weights of network NET^S. The information learned from the source domain can be transferred to the destination by using $\mathcal{W}^S$ as the initial values of the trainable weights for NET^D. By way of explanation, NET^D leverages $\{\mathcal{W}^S, \mathbb{D}^D\}$ to optimize its nonlinear transformation functions.

Although the weights $\mathcal{W}^S$ do not contain information about the propagation environment in the destination domain, they are enough for the network to predict the beam pair corresponding to the LOS, which is a non-linear function. By transferring the trained weights $\mathcal{W}^S$ to NET^D, having to re-learn such mapping in the destination domain is avoided. The Network NET^D can tune its parameters to consider the NLOS cases in the destination environment with few measurements. As the input and output spaces are the same for both NET^S and NET^D, the TL is inductive and homogeneous [9].

2) Task Adaptation: Collecting many measurements of the environment with all possible antenna configurations for TX and RX is clearly infeasible. Using TL, the knowledge about a given propagation environment that is present in a dataset obtained with a specific antenna configuration can be partly reused in other antenna setups. The hypothesis is that, by using this technique, only a small training dataset with the new antenna configuration is required to obtain acceptable performance. We denote with NET^B and NET^T the beam selection networks at the same environment with the base and target antenna configurations of transceivers $\{N_t^B; N_r^B\}$ and $\{N_t^T; N_r^T\}$, respectively. Both networks have exactly the same structure and number of neurons except for their output layers, whose dimensions follow their specific antenna configurations: $N_t^B N_r^B$ neurons in NET^B and $N_t^T N_r^T$ in NET^T.

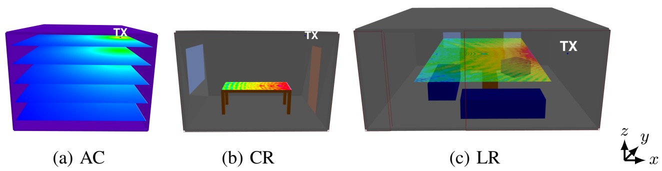
Assume the network NET^B is accurately trained due to the large training dataset $\mathbb{D}^B$, and it reliably provides the best beam pair at the base configuration. Since both networks use samples from the same environment and the first layers of a deep neural network can be seen as a feature extraction module [12], the weights corresponding to the initial layers of NET^B can be used as initialization of the corresponding weights at NET^T. Thus, the learned mapping from position and orientation of RX to appropriate directions can be transferred. Since the structure of both networks is the same for their input and hidden layers, we use the trained weights for the base antenna configuration $\mathbf{W}_1^B, \mathbf{W}_2^B, \dots \mathbf{W}_{N_h}^B$ as initial values for the weights $\mathbf{W}_1^T, \mathbf{W}_2^T, \dots \mathbf{W}_{N_h}^T$ in the target configuration network NET^T. As the output layers have different dimensions in both networks, we use random initialization of the weights $\mathbf{W}_O^B$ at the output layer of NET^T. Weights of NET^T are then fine-tuned with a dataset obtained with the target antenna configuration. Contrary to domain adaptation, the learned mapping between coordinate and beam indices in LOS and NLOS situations is transferred partially to NET^T. As the output spaces in NET^B and NET^T are different, the TL is categorized as inductive and heterogeneous [9].

3) Freezing Trainable Parameters: The hidden layers of the network can be seen as the part learning the most common directions for each input $\mathbf{x}$, while the output layer maps these directions to the right beamforming codewords. With this interpretation, as in the domain adaptation the codebook is the same, we have the option to freeze the output layer after loading $\mathbf{W}_O^S$. In the task adaptation case the environment is the same, so the weights of the hidden layers can be frozen after loading $\mathbf{W}_1^B, \mathbf{W}_2^B, \dots \mathbf{W}_{N_h}^B$, with only the last layer being fine-tuned. Freezing these layers decreases the number of trainable parameters in the network, which improves performance especially when few samples are available for training.

IV. Simulation Results and Discussions

We present numerical evaluations to compare the proposed TL based beam selection with two baselines: the DL-based method without TL [7] and hierarchical beam-search [1]. We consider a 3D scenario with three different environments, an anechoic chamber (AC), a conference room (CR), and a living room (LR), which are shown in Fig. 2. The AC represents an environment without any scatterers, where only the LOS path is present in the wireless channel. The CR and LR are described in detail by IEEE 802.11ad task group [11], and their datasets are generated with half of the instances in LOS conditions and the other half in NLOS conditions. The main information about the indoor environments is summarized in Table I. The different indoor environments are used to assess the TL idea in the domain adaptation case.

We use the accurate ray-tracing tool, Altair Feko-Winprop software [13] to generate channel responses. The ray tracing tool provides all the necessary information such as the angle of departure (AoD), angle of arrival (AoA), path gains, etc. for all paths to construct accurately the channel response between the TX and RX using (1). In the ray-tracing tool, empirical losses for transmission, reflection and diffraction are considered. We



(a) AC (b) CR (c) LR

Fig. 2. Three standard indoor environments where the rooms have different dimensions and static objects. The LOS power is illustrated for the user grids of the environments.

TABLE I. Geometrical information of the indoor environments

	Dimension (m)	AP's pos. (m)	User grid (m)	Dataset size
AC	$4 \times 5 \times 3$	$(3.5, 2.5, 2.5)$	$[0-3.5] \times [0-5] \times [0.5-2.5]$	90,000
CR	$4 \times 3 \times 3$	$(4, 1.5, 2.9)$	$[1-3.5] \times [1-2] \times [1]$	6,250
LR	$7 \times 7 \times 3$	$(7, 3.5, 1.5)$	$[1.5-5.5] \times [0-7] \times [1.5]$	70,000

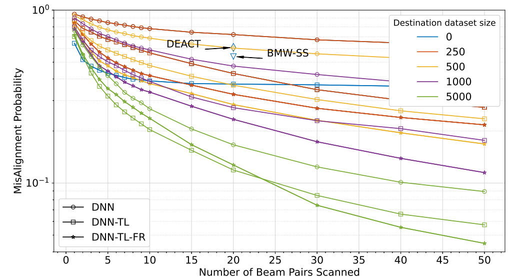
use the 25 strongest paths at each receiver location to construct the channel response.

The parameters $P_t = 0\,\mathrm{dBm}$ and $\sigma_n^2 = -84\,\mathrm{dBm}$ are used in the dataset generation step. After generating the datasets by calculating the RSS for each beam pair at each user position, 80% of each dataset is used to train a network and the rest for the evaluation process. Thus, there are $72,000$, $5,000$, and $56,000$ samples for training in the AC, CR, and LR, respectively. On the structure of all neural networks, we consider 5 hidden layers with 128 neurons at each layer and 10% dropout for all hidden layers to prevent overfitting. Adam optimizer with 50 epochs is used to train the neural network, with a minibatch size gradually increasing from 32 to 8192 samples [7]. Also, the labels are converted to one-hot vectors which are used in the calculation of the cross entropy as loss function. To replicate the simulation results, the code and datasets are available in https://github.com/SajadRezaie/DeepTLBeamSelection. In the evaluations, we consider three training-based approaches:

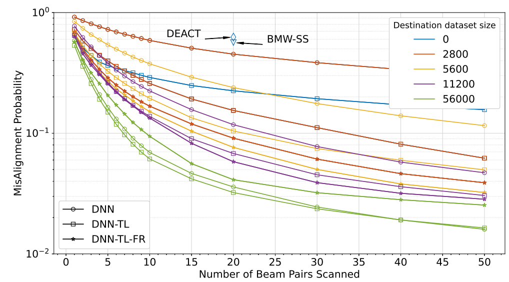
1) **DNN**: The DNN method with random weight initialization and trained on the destination/target datasets,
2) **DNN-TL**: The DNN method in which source weights are used as initialization, and all layers are fine tuned with the destination/target dataset,
3) **DNN-TL-FR**: The DNN method in which source weights are used as initialization, and some layers (the output layer and hidden layers, respectively, in domain and task adaptation) are frozen and the rest are fined tuned witht the destination/target dataset.

To assess the effects of the TL technique in the domain adaptation scenario, we consider 64 and 16 antenna elements, respectively, at the TX and RX of the 3 environments. We train the network NET^S in the AC as source domain, with a large training dataset $\mathbb{D}^S$ including $72,000$ realizations. For fine-tuning of the network NET^D at the destination domain CR or LR, we use a subset of the training dataset to evaluate the impacts of the destination dataset size on the TL performance.

Fig. 3 shows the misalignment probability, i.e. the probability of missing the best beam pair in a candidate list with limited size N_b, of different beam selection methods by processing only a portion of the destination dataset at the CR or LR. We consider both of the CR or LR as a destination domain to evaluate the effects of different propagation properties and



(a) Destination domain: the CR



(b) Destination domain: the LR

Fig. 3. Misalignment probability of DL-based beam alignment with and without using TL for domain adaptation. The full destination training dataset includes $5,000$ and $56,000$ samples at the CR and LR, respectively.

different dataset sizes at the TL process. The plots with 0 samples depict the case where no samples captured at the destination environment are used, and there is no fine-tuning in the TL method. We also show the beam selection method's performance using the full datasets to have a lower bound for comparison. Besides, the performance of two hierarchical beam-search methods (DEACT and BMW-SS) from [1] is shown, where the number of beam pair scans needed by these methods is fixed and fully determined by the antenna array configurations (20 in this case).

As illustrated in Fig. 3, when training with a very limited dataset (5% of the training dataset) a significant improvement in the beam alignment is achievable using the TL technique. Furthermore, by freezing the output layer with 132096 parameters, the number of trainable parameters decreases from 198784 to 66688. Freezing around 66% of the weights helped the network to work better especially with very few training samples in the destination domain. As shown in Fig. 3, the proposed beam selection method using the TL technique outperforms both the DEACT and BMW-SS methods. The results show transferring only the LOS information from the source domain AC can be very useful.

Fig. 4 shows the achievable spectral efficiency using hierarchical and DL-based beamforming. By assuming a fixed channel coherence time, the proportion of channel resources used for beam alignment for each coherence time period is subtracted from the system's achievable rate. Hence, the effective spectral efficiency can be defined as

$$\mathrm{SE}_{\mathrm{eff}} = \frac{T_{fr} - N_b T_s}{T_{fr}} \log_2(1 + \mathrm{SNR}_{p,q}), \quad (4)$$

where T_s and T_{fr} denote the time required to scan a beam pair in beam alignment process and the time duration of one frame with fixed channel response, respectively [14]. Also, $\mathrm{SNR}_{p,q}$

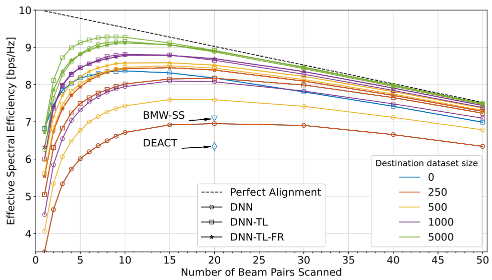


Fig. 4. Spectral efficiency of different beam alignment at the CR, with the option of transferring learned knowledge from the AC.

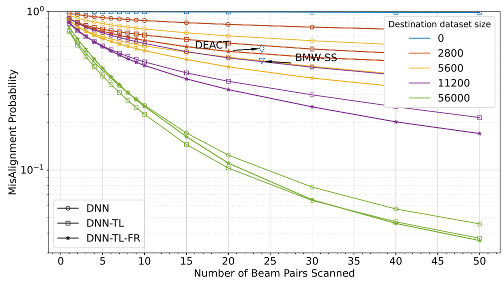


Fig. 5. The different beam alignment performance at the LR with antenna configuration of transceivers $\{64; 64\}$, where the learned knowledge from the same environment with $\{64; 16\}$ antenna elements can be transferred.

is the SNR corresponding to the selected beam pair (p, q) after the beam alignment phase, and is defined as

$$\mathrm{SNR}_{p,q} = \frac{\left\| \sqrt{P_t} \boldsymbol{v}_q^H \boldsymbol{H} \boldsymbol{u}_p s \right\|^2}{\sigma_n^2}. \quad (5)$$

In the simulations, we consider $T_{fr} = 20$ms and $T_s = 0.1$ms [14]. Contrary to HBS, in which the number of scanned beam pairs is fixed, N_b can be tuned in the DL methods by adjusting the candidate beam list size. By using TL and freezing the output layer, DL-based beamforming provides the same spectral efficiency as HBS methods with only 3 scanned beams, hence reducing the alignment latency by 85%.

The capability of TL in the task adaptation problem is shown in Fig. 5, where the tasks are beam alignment with different antenna configurations in a same environment. At the LR, a large dataset is captured with TX and RX having 64 and 16 antenna elements, respectively. In the same environment, we collect a small dataset with antenna configuration of transceivers $\{64; 64\}$. In line with our expectation, increasing the dataset size lowers the misalignment probability, and using TL proves to be more useful with more limited datasets. In addition, Fig. 6 shows the performance of different beam alignment methods in the reverse case, where the target task is alignment of transceivers with $\{64; 16\}$ antenna elements, and we can transfer the trained weights with $\{64; 64\}$ antenna elements. Since beam alignment with antenna configuration $\{64; 16\}$ is easier than $\{64; 64\}$, smaller target dataset sizes are needed to obtain acceptable performance. As illustrated in Fig. 6, freezing the hidden layers helps to have better performance, but not as much as the domain adaptation case. The reason is that here we freeze only around 34% of the parameters. These improvements prove the potential of using the TL technique

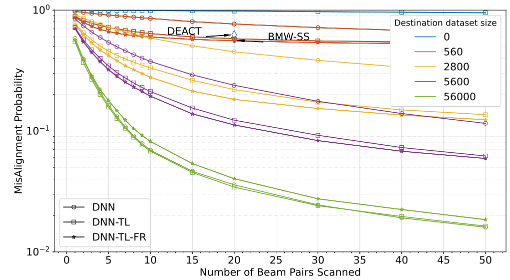


Fig. 6. Misalignment probability of the beam alignment methods at the LR with antenna configuration of transceivers $\{64; 16\}$, with weights being transferred from the same environment with configuration $\{64; 64\}$.

across neural networks of different dimensions, even when the output layer is initialized with random weights.

V. Conclusions

The findings of this paper have shown that transfer learning techniques, in particular parameter reuse, can be leveraged to significantly reduce the training data requirements for deep neural networks performing location- and orientation-based beam alignment. Trained neural network weights can be effectively reused in other propagation environments or antenna configurations by fine tuning them with small datasets in the destination environment or the target antenna setup. Future research will focus on exploring more sophisticated transfer learning techniques, and extending the deep learning paradigm to location-aware beam tracking.

References

[1] Z. Xiao, T. He, P. Xia and X. Xia, "Hierarchical Codebook Design for Beamforming Training in Millimeter-Wave Communication," *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3380–3392, May 2016.

[2] M. Kok, J. D. Hol and T. B. Schön, "Using Inertial Sensors for Position and Orientation Estimation", *Found. Trends Signal Process.*, vol. 11, nos. 1-2, pp. 1–153, 2017.

[3] V. Va, J. Choi, T. Shimizu, G. Bansal and R. W. Heath, "Inverse multipath fingerprinting for millimeter wave V2I beam alignment," *IEEE Trans. Veh. Technol.*, vol. 67, pp. 4042–4058, May 2018.

[4] M. Alrabeiah and A. Alkhateeb, "Deep Learning for mmWave Beam and Blockage Prediction Using Sub-6GHz Channels," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5504–5518, Sep. 2020.

[5] W. Xu, F. Gao, S. Jin and A. Alkhateeb, "3D Scene Based Beam Selection for mmWave Communications," *IEEE Commun. Lett.*, 2020.

[6] Z. Zhu and K. Lin, and J. Zhou, "Transfer Learning in Deep Reinforcement Learning: A Survey," *arXiv*:2009.07888, Sep. 2020, [Online] Available: https://arxiv.org/abs/2009.07888.

[7] S. Rezaie, C. N. Manchón, and E. D. Carvalho, "Location- and Orientation-Aided Millimeter Wave Beam Selection Using Deep Learning," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2020, pp. 1–6.

[8] C. Lin, W. Kao, S. Zhan and T. Lee, "BsNet: A Deep Learning-Based Beam Selection Method for mmWave Communications," in *Proc. IEEE 90th Veh. Technol. Conf.*, Sep. 2019, pp. 1–6.

[9] F. Zhuang et al. "A Comprehensive Survey on Transfer Learning," in *Proc. of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.

[10] W. Alves, I. Correa, N. González-Prelcic and A. Klautau, "Deep Transfer Learning for Site-Specific Channel Estimation in Low-Resolution mmWave MIMO," *IEEE Wireless Commun. Lett.*, Mar. 2021.

[11] A. Maltsev et al., "Channel models for 60 GHz WLAN systems," *IEEE Document*, 802.11-09-0334r6, Jan. 2010.

[12] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Mar. 2013.

[13] Altair Feko Winprop, https://www.altair.com/feko-applications.

[14] M. Hussain and N. Michelusi, "Second-Best Beam-Alignment via Bayesian Multi-Armed Bandits," in *Proc. IEEE GLOBECOM*, Dec. 2019, pp. 1–6.