Towards Virtual to Real-world Transfer Learning for Mobile mmWave Beam Tracking

Christos Laskos, Sigrid Dimce, Anatolij Zubow and Falko Dressler
Technische Universität Berlin, Germany
{laskos, dimce, zubow, dressler}@tkn.tu-berlin.de

***Abstract*—Adaptive beamforming is an enabling technology for millimeter-wave-based wireless communication which is used by many standards like 3GPP NR and IEEE 802.11ay. Supporting user mobility is challenging as efficient beam tracking is required. Therefore, a variety of beam tracking techniques have been proposed, many of which are machine learning (ML)-based. However, ML approaches are often not of practical use due to the long and complex learning phase. In this paper, we show the feasibility of virtual to real-world transfer learning. Our solution significantly speeds up the learning process as the learning happens mostly in a simulated environment and requires only little additional learning in the real-world deployment. As proof-of-concept, we implemented and evaluated a low-complexity beam tracking based on deep Q-network (DQN) reinforcement learning. The results reveal a substantial speed-up by a factor of $3\times$ using transfer learning.**

***Index Terms*—mmWave, beam tracking, machine learning, transfer learning, deep Q-network, software defined radio**

I. Introduction

In recent years, millimeter wave (mmWave) communications has gained a lot of attention for the use in next-generation communication systems like 5G NR, promising high data rates and low latency communication [1], [2]. This is achieved through the use of large bandwidths and high carrier frequencies of 24 GHz and higher – in this paper, we focus on the 60 GHz ISM band. Unfortunately, the high frequencies in the mmWave band possess higher attenuation and are more susceptible to being negatively impacted by obstacles such as vegetation, humans, and weather conditions. Additionally, the license-free band around 60 GHz suffers a very high attenuation caused by the oxygen molecules in the atmosphere and rain drops [3].

To counter the higher attenuation, electronic beam forming (EBF) with phased array antennas is used [4]. These antennas increase the signal power by focusing the transmitted signal in a specific direction, i.e. location of the mobile station. Such a created beam has a specific width, depending on the design of the antenna, e.g., number of antenna elements [4]. The limited beam width leads to the situation where the mobile user eventually moves out of the beam, and the signal degrades. To prevent this, beam tracking is essential, in which the beam direction is adapted in real-time according to the movement of the user.

Current approaches for beam tracking can be classified into pure data driven approaches [5] and those using machine learning (ML) techniques [6]–[9]. Although ML approaches provide better performance, they are often not practical because of their long and complex learning phase.

In this paper, we show the feasibility of virtual to real-world transfer learning (TL) which dramatically speeds up the learning process as the learning happens mostly in a simulated environment and requiring only little additional learning in the real-world deployment. We use TL to transfer a baseline beam tracking agent from virtual to real environment. We apply TL by changing the observation to real-world input and performing a short retraining period. With the training completed, the agent can then be deployed at the base station (BS) and track the mobile station (MS) with the same performance as an agent trained purely in the real environment, requiring only 1/3 the training time. At last, we test both TL and non TL cases in a high noise environment and observe a slightly higher performance of the TL agent, indicating a higher noise robustness than the conventional agent.

This is demonstrated using the example of mmWave beam tracking based on deep reinforcement learning (RL). We specifically choose a deep Q-network (DQN) for the application of TL to beam tracking because of the recent successful application of a DQN to mmWave beam tracking [9]. We propose a low-complexity system, using only easily available data from the MS, such as a smartphone, and remove the need for complex channel estimation and transmission of pilot signals. Our approach, is first verified in simulation environment using the *Phased Array System Toolbox* provided by MATLAB. Afterwards, the trained agent is used as a baseline for the application of TL.

Our main contributions can be summarized as follows:

- We present a low-complexity mmWave beam tracking approach based on DQN reinforcement learning,
- We show the feasibility of virtual to real-world transfer learning using the example of mmWave beam tracking,
- Results from experimental reveal that an agent using TL achieves same performance but in a much shorter learning time. Moreover, it can be applied in scenarios with high observation noise.

II. Related Work

Some applications of TL in wireless networks include spectrum sensing, channel selection, and channel estimation [10]. In spectrum sensing, Zheng et al. [11] pretrain a convolutional neural network with many different signal types and adapt the pre-trained network to real-world signals. With the usage of TL, their approach shows higher detection performance in low SNR conditions compared to non TL and traditional spectrum sensing methods. Using TL in channel selection, Lin et al. [12] transfer knowledge between different frequency bands and improve prediction accuracy when training data is limited. Parera et al. [13] applied TL to channel estimation and transfer knowledge between LTE channels for channel quality prediction. Using TL in this scenario, shows improved performance when the available data from the target channel is limited.

We apply TL to beam tracking. The first beam tracking algorithms were based on extended Kalman filter (EKF). In an early work, Zhang et al. [14] suggest to perform a full scan of all available beams to create an observation matrix, which is used as input to the EKF for beam selection. Extending this work, Va et al. [15] increased the efficiency of the system by removing the need for a full scan, allowing for faster beam switching in mobile environments, using only one measurement. Further improvements were made by Jayaprakasam et al. [16], showing better performance and eliminating the need for channel re-acquisition. Liu et al. [17] remove the need for prior knowledge of unknown path gains and start by using two well-configured tracking beam pairs. A beam tracking system based on a particle filter was proposed by Lim et al. [18]. In their work, they perform the alignment and tracking using pilot signals transmitted to the receiver. A data-driven beam tracking approach was developed by Ma et al. [5], which uses the signal strength as the tracking criterion, which has to surpass a predefined QoS threshold for a beam to be selected.

More recent approaches rely on deep learning. Ma et al. [8] tested convolution neural networks and long short-term memory as training networks for improved beam training, which is the initial step to acquire the starting beam configuration. They were able to reduce the overhead of conventional sweeping approaches and increase the alignment accuracy. An actual beam tracking implementation based on deep learning was developed by Lim et al. [6] using a long-short term memory (LSTM) prediction model to predict the next beam based on previous observations. They use channel state information that is collected during specific beam transmission periods, where some beams are selected and the channel is estimated. A similar approach was developed by Liu et al. [7], which leverages a model-based approach using traditional signal processing with a matched filter that is then used as the input to their deep learning neural network. Park et al. [9] propose to use a DQN. Here the received pilot signals are used as inputs to the neural network, which then selects the best-suited beam.

This approach also inspired us to evaluate DQN beam tracking with transfer learning. We first simplify the required inputs to the DQN by only using easily available data from the MS. Furthermore, we take the trained agent from virtual to real-world environment and evaluate TL and non TL cases in the real world. At last, we introduce noise to the environment and evaluate the robustness of both cases in the real environment. To the best of our knowledge, TL has not yet been applied to beam tracking.

III. Background

A. mmWave Beamforming

Beamforming uses the concept of electromagnetic interference to improve wireless connectivity by focusing the signal toward a specific receiving device [19]. This is achieved by combining the elements of an antenna in such a way, that in certain directions the signals interfere constructively, while in other directions interference is destructive. Beamforming is usually an electronic process, where the beam can be re-calibrated electronically to aim in a new direction so that the beam is aligned towards the intended MS. Here, the usage of EBF is of paramount importance in order to overcome the high signal attenuation of the mmWave spectrum. However, in order to make practical solutions inexpensive the EBF cannot be freely controlled. Instead, standard beam-training algorithms are used, which probe a set of pre-defined antenna patterns and select the best configuration with respect to metrics like signal strength or SNR [20]. Note, that sweeping through all predefined antenna sectors is in general not feasible due to the high signaling overhead. Therefore, especially in mobile environments, techniques like beam tracking are applied, where the EBF is constantly adjusted to maintain communication towards the MS. Failing to adjust the configuration of EBF can result in a link outage and may require costly sweeping to reacquire the correct beam configuration.

With the IEEE 802.11ad/ay (WiGig) standard [21], WiFi is able to use the mmWave spectrum at 60 GHz. As an example, a commodity WiGig AP like the Talon AD7200 uses a beam pattern set with 35 configurations [22].

B. Virtual to Real-World Transfer Learning

Transfer learning (TL) is a technique in ML in which knowledge learned from one task is re-used in order to boost performance on a related task [23]. As an example consider the task of image classification where knowledge gained while learning to recognize dogs could be utilized when trying to recognize cats. The key idea behind transferring knowledge from previously learned tasks to new tasks is to accelerate the learning speed and therefore efficiency.

The application of ML techniques in networking in general and beam tracking in particular is very promising as it is able to provide superior performance over classically designed approaches [23]. However, they are often not practical because of their long and complex learning phase requiring large amount of adequate training data. This is where the use of TL could prove useful. Adequate training data can be collected using simulation-based (virtual) training at a much lower cost and higher learning speed, and the trained model can be transferred to the real physical system, e.g., robot [24] or a cellular BS, see Fig. 1. However, the transfer of knowledge from virtual/simulated environments to the real world often encounters a fundamental mismatch, but this challenge can be effectively addressed by providing sufficient realistic virtual environments. In the sense of TL, the tasks mainly differ in their degree of realism.

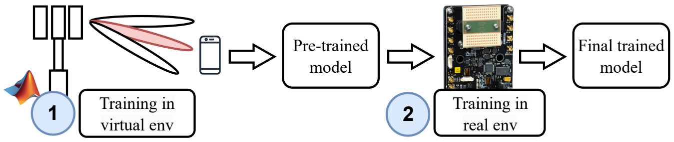


Fig. 1. Virtual to real-world transfer learning.

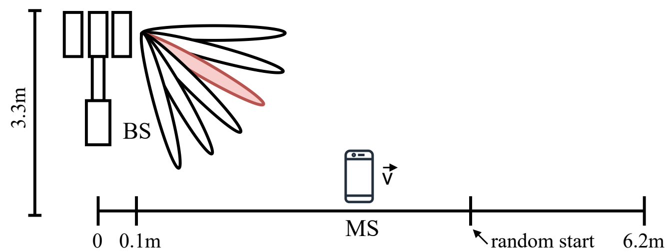


Fig. 2. Scenario under study with MS starting at random position and moving along a straight line with random speed/direction.

IV. System Model & Problem Statement

We consider a network consisting of a single BS serving a MS, see Fig. 2. The BS is equipped with a phased array antenna operating in mmWave spectrum with K number of different beam configurations. We consider the downlink (DL) of a time-slotted transmission scheme. The beam configuration of the BS can be adapted at the beginning of each time slot. The MS is mobile and moves along a straight line with random speed and into random direction (left or right). Speed is randomly selected between 1 or 2 m/s. Moreover, its start position is selected randomly. To make the mobility pattern more realistic, the mobile agent starts on either end of the path and traverses it until the other end. Additionally, with 25% probability he can change direction in the middle of the path.

The objective of the BS is to select the beam configuration out of the K possible for the next time slot which maximizes the signal strength at the MS.

V. DeepMMTrack Approach

A. Description

`DeepMMTrack` is a low-complexity mmWave beam tracking approach based on deep RL which is described in this section. The `DeepMMTrack` architecture is shown in Fig. 3. For the DQN agent, we used the implementation provided by Matlab[1]. We define observation, action, and reward as follows:

Observation: The observation consists of two parts. First, a vector of receive power values of selected beam configurations in the last N time slots $\mathrm{Prx}_b^{N\ldots1}$. Second, the speed and movement direction of the MS.

Action: The beam configuration selected by the BS for the next time slot.

Reward: The reward function is defined as:

$$r = \begin{cases} \mathrm{Prx}_b^t & \text{if } \mathrm{Prx}_b^t \geq 5 \\ -10, & \text{otherwise} \end{cases} \tag{1}$$

[1]Matlab *rlDQNAgent* reinforcement learning agent

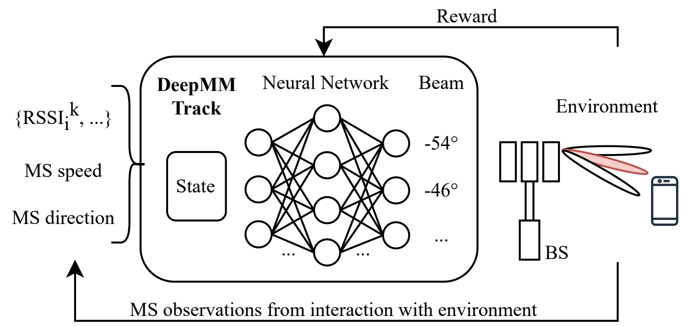


Fig. 3. `DeepMMTrack` architecture: RL agent residing in the BS uses DQN to decide on the next beamforming configuration using the observations made by the MS.

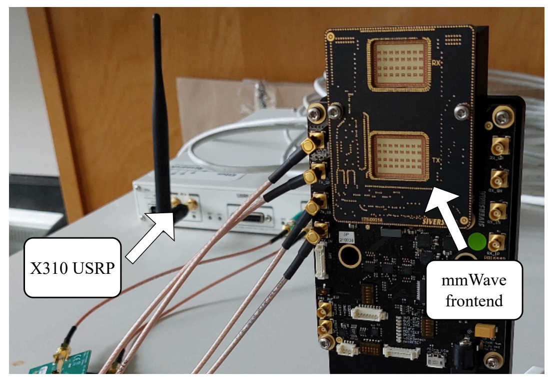


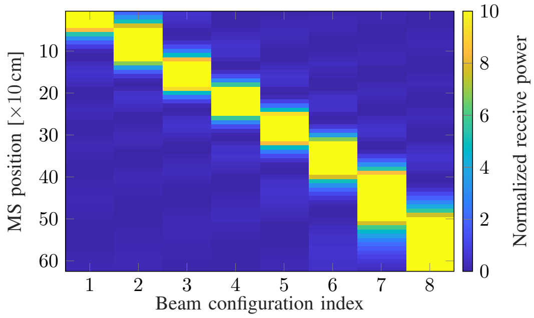
Fig. 4. Prototype of BS with mmWave frontend and signal processing using an SDR.

where Prx_b^t is the normalized receive power at the MS in the last time slot t and selected beam configuration b. Hence the agent can receive rewards between 5 and 10, as well as negative punishment of -10. The negative punishment is used to accelerate training and minimize the selection of wrong beam configurations.

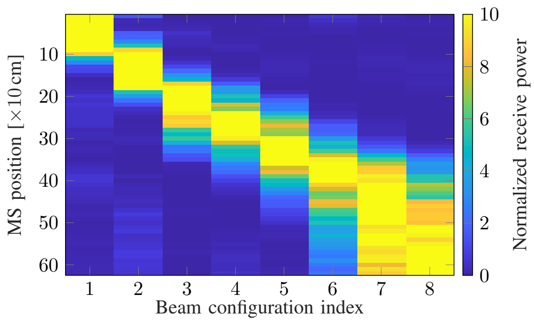
We use a deep neural network with one fully connected layer, followed by a dropout layer, a rectified linear unit (ReLU) activation layer, and lastly the output. The dropout layer has a 10% chance of dropping neurons and is used to prevent over-fitting.

B. Prototype

We prototypically implemented `DeepMMTrack` using commercially available mmWave frontends (Sivers EVK06005 mmWave front-end with BFM06009 antenna) whereas the baseband processing was performed by software-defined radio (SDR) Ettus USRP X310 (Fig. 4). The mmWave frontend produces beams with a width of 12°, that are steerable from $-54°$ to $+54°$ in 5.4° steps. The carrier frequency was set to 60.48 GHz. As waveform we selected IEEE 802.11a OFDM which was generated and processed using Matlab. The BS is placed in one corner of our lab, while the MS is mounted on a robot used for alignment, which is then moved on a tray.



(a) Virtual (simulated) environment.



(b) Real-world (lab) environment.

Fig. 5. Normalized receive power for different MS positions and beam configurations.

VI. Evaluation

A. Methodology

In order to show the feasibility of virtual to real-world transfer learning using the example of mmWave beam tracking we set up two environments, one virtual and one real.

1) Real Environment: We recreated the scenario from Fig. 2 in our laboratory and used the prototype described in Section V-B. We placed the MS at each of the 62 positions, which is the maximum distance reachable in our lab, and measured the signal strength for each beam configuration (cf. Fig. 5b). We transmitted 802.11a (BPSK, 20 MHz) WiFi frames from the BS to the MS. We chose small bandwidth 802.11a frames because we are only interested in the signal strength information, larger bandwidth and high data rates are not required in our beam tracking scenario. The actual learning of the agent occurred by playing back this trace while taking into account the speed and direction of the MS.

2) Virtual Environment: The virtual (simulated) environment mimics the real one described in previous section. We modeled all relevant aspects like the geometry of the mmWave phased array as well as the signal path loss in the simulation. Specifically, we used the MATLAB *Phased Array System Toolbox*. The measured signal strength for each position and beam configuration is shown in Fig. 5a. Note, the large similarities but also differences as compared to the results obtained in the real environment (cf. Fig. 5b). The latter is due to the objects such as chairs and tables not taken into account in the simulation.

3) Modelling Noise: In order to analyze the impact of noisy observations in the measured signal strength, i.e., RSSI_i, the following model is used:

$$Y = X + w \tag{2}$$

where X is the ground truth observation matrix and w is a random variable representing Gaussian noise with zero mean and some standard deviation σ. Hence the observation available to the agent is Y instead of the noise-free X.

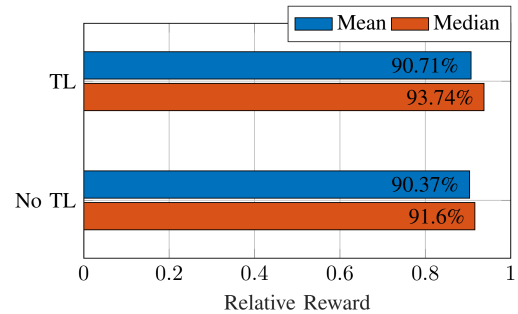


Fig. 6. TL vs. no TL reward after 1000 test runs.

B. Hyperparameter Estimation

We optimized the hyperparameters for `DeepMMTrack` by testing different lengths for signal strength history (sh), neural network sizes (nn), and learning rates (lr). Using the simulation environment, we tested 180 different combinations of hyperparameters and evaluated their performance in 1000 test runs. Based on these results, the agents were scored according to their mean reward and their reward deviation. A top performing agent requires a high reward and a low reward deviation. Best results were obtained with $sh = 16$, $nn = 3$, $lr = 1 \times 10^{-5}$.

C. Feasibility of Transfer Learning

We evaluated the performance of `DeepMMTrack` with and without TL. Here we are interested in investigating whether or not the TL has an influence on the final performance after learning. We start with optimal conditions, absence of any noise in the observation, i.e., $\sigma = 0$. Fig. 6 shows the mean and median relative reward computed over 1000 test runs and 25 different seed values. We observe that both cases have similar performance, with nearly identical mean performance, but the agent with TL has a 2% higher median. Both agents perform very well in the scenario and are suitable for beam tracking.

Next, we analyze the learning speed of both approaches. Fig. 7 shows the relative reward for each training episode of both approaches. To filter out the volatile reward per episode, we show the moving average over 100 training episodes. We can observe the much faster learning when using TL. The agent with TL starts with a much higher reward, starting at about zero, whereas the agent without TL lacks behind with a start reward of -0.5. Additionally, the agent using TL completes the training process significantly faster than the agent without. Both agents have a stop training criterion when reaching 90% of the maximum reward over the last 40 episodes. The TL agent has completed its training in 6100 episodes whereas the

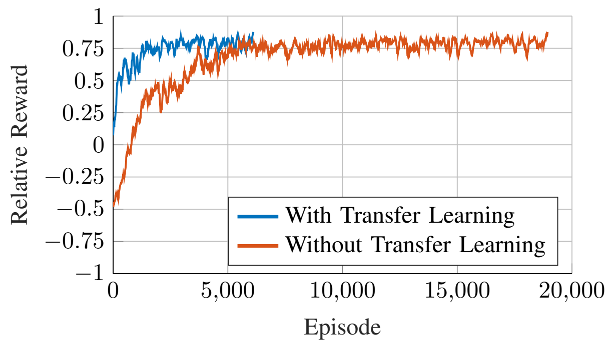


Fig. 7. TL vs. no TL training process.

agent without transfer required 19 000 episodes. In our case, TL can reduce the required training time to just 32 %.

Our results confirm the feasibility of virtual to real-world TL where an agent trained purely in a simulated environment can quickly adapt to a real-world environment. The same performance is achieved, but in much less learning time. The simplifications made in the virtual/simulated environment had no visible impact.

D. Impact of Noisy Observation

Finally, we analyzed the impact of a noisy observation on the performance of the agent (Section VI-A3). For this we have analyzed different levels of noise on the observed signal strength, i.e. $\sigma = [1, 2, 3]$ (Eq. (2)), and compared it to a perfect environment without any noise, i.e., $\sigma = 0$. First, we examine the performance of a conventional agent without TL in the lab environment. From Fig. 8a, we can observe that with an increasing σ the agent performance drops. The achieved mean relative reward at $\sigma = 3$ is $3\times$ lower compared to the environment without noise. However, even with highest noise, resulting in frequent punishments, the agent is able to track the mobile user without completely losing its performance, i.e. relative reward remains > 0 where the agent receives more rewards than punishments. When applying TL, as shown in Fig. 8b, performance loss is similar to the conventional learning case, with TL performing marginally better, especially in high noise conditions. Comparing both cases, we can conclude that TL may additionally offer higher robustness against noise.

VII. Discussion

Next, we discuss the advantages, limitations, and future work we plan for our `DeepMMTrack` system.

A. Advantages

With the introduction of `DeepMMTrack`, beam tracking can be successfully transferred from virtual to real environment. This enables purely simulation trained agents to be adapted to real-world scenarios with reduced training amount. Compared to the conventional training case, TL offers a training time reduction of 68 %. Additionally, in the case of noisy environments, TL shows more robustness with higher tracking performance than the conventional case. Furthermore, we are able to simplify the required inputs to the RL agent while maintaining high tracking performance, both in virtual and real environments. Overall, `DeepMMTrack` enables the transfer of beam tracking from virtual to real environment with minimal training effort, an increase in noise robustness compared to the conventional case, and a simplification of the required tracking inputs.

B. Limitations

1) Multi-user Beam Tracking: So far, we considered the single-user beam tracking. However, `DeepMMTrack` can be used to track multiple users for which no modifications need to be made. This is because all relevant information is contained in the observation space. Moreover, it enables new MS to join the network in a smooth way since the RL agent is already trained.

2) Realistic Mobility Models: Currently, we modeled user mobility only in 1D, limiting the possible movements of the user. To accurately model user mobility, more complex and hence realistic mobility models are required. Specifically, examining MS mobility in 2D is necessary. Even 3D space might be considered, which would make the use of 2D beam steering necessary.

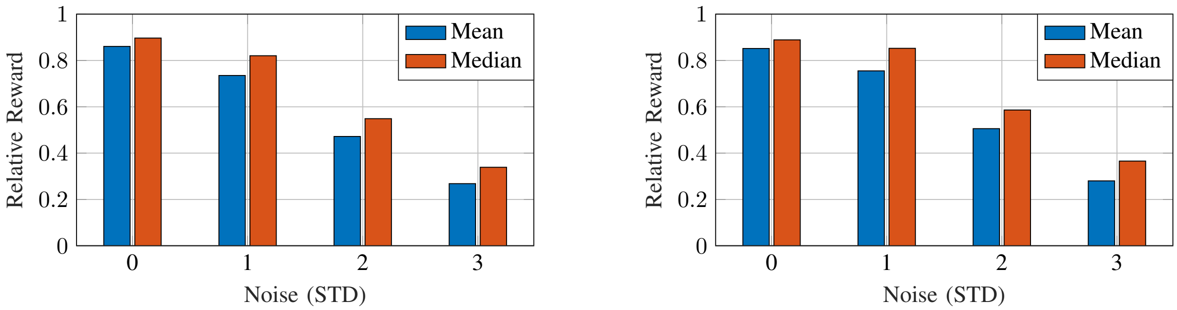
3) Influence by External Factors: So far, we have assumed optimal conditions for mmWave communication. However, in reality the line of sight path might be blocked by objects like trees. Moreover, the mmWave signal might be attenuated by rain or snow.

C. Future Work

In future work, we plan to extend `DeepMMTrack` for tracking multiple users simultaneously starting in 1D and later extending to more complex environments with 2D movements. We may also consider 3D movements to emulate mobile users in an area with many elevation changes. Another extension we plan, is the realistic modeling of mmWave channel fading. Currently, we only considered Gaussian noise and plan to extend the noise modeling to real-world conditions by performing measurements in our lab over a long period of time. Additionally, we plan to investigate the impact of external factors on beam tracking performance, such as rain or snow, and create a more realistic environment. With these extensions, we aim to build a transferable beam tracking system, which closely models real-world environments.

VIII. Conclusion

In this paper, we studied the use of TL for use in beam tracking. TL promises many advantages such as a significantly shortened training time after deployment. We first simplify the required inputs of DQN mmWave beam tracking using only readily available data from the MS. After successfully evaluating our approach in the simulated environment, we apply transfer learning to evaluate real-world performance. We are able to show that virtual to real-world transfer learning is not just feasible for mmWave beam tracking. Results from experiments reveal that an agent using transfer learning achieves same performance but only requires 1/3 of learning time.



(a) Without TL.

(b) With TL.

Fig. 8. Noisy lab environment.

Moreover, pre-learning in a simulated environment is much more cost-effective because it is not limited by the actual speed of a robot or human carrying the MS. With our contribution, we lay the foundation for transferring beam tracking from purely simulation based evaluation to real-world prototype with minimal additional effort.

Acknowledgements

This work was supported by the Federal Ministry of Education and Research (BMBF, Germany) within the 6G Research and Innovation Cluster 6G-RIC under Grant 16KISK020K as well as by the German Research Foundation (DFG) within the project ML4WiFi under grant DR 639/28-1.

References

[1] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, and G. Chen, "Millimeter Wave Communication: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1616–1653, 2018.

[2] S. Kutty and D. Sen, "Beamforming for Millimeter Wave Communications: An Inclusive Survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 949–973, 2016.

[3] F. Giannetti, M. Luise, and R. Reggiannini, "Mobile and personal communications in the 60 GHz band: A survey," *Wireless Personal Communications*, vol. 10, pp. 207–243, 1999.

[4] S. A. Busari, K. M. S. Huq, S. Mumtaz, L. Dai, and J. Rodriguez, "Millimeter-Wave Massive MIMO Communication for Future Wireless Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 836–869, 2018.

[5] Y. Ma, S. Ren, W. Chen, and Z. Quan, "Data-Driven Beam Tracking for Mobile Millimeter-Wave Communication Systems Without Channel Estimation," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2747–2751, Dec. 2021.

[6] S. H. Lim, S. Kim, B. Shim, and J. W. Choi, "Deep Learning-Based Beam Tracking for Millimeter-Wave Communications Under Mobility," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7458–7469, Nov. 2021.

[7] J. Liu, X. Li, T. Fan, S. Lv, and M. Shi, "Model-Driven Deep Learning Assisted Beam Tracking for Millimeter-Wave Systems," *IEEE Communications Letters*, vol. 26, no. 10, pp. 2345–2349, Oct. 2022.

[8] K. Ma, D. He, H. Sun, Z. Wang, and S. Chen, "Deep Learning Assisted Calibrated Beam Training for Millimeter-Wave Communication Systems," *IEEE Transactions on Communications*, vol. 69, no. 10, pp. 6706–6721, Oct. 2021.

[9] H. Park, J. Kang, S. Lee, J. W. Choi, and S. Kim, "Deep Q-Network Based Beam Tracking for Mobile Millimeter-Wave Communications," *IEEE Transactions on Wireless Communications*, vol. 22, no. 2, pp. 961–971, Feb. 2023.

[10] C. T. Nguyen, N. Van Huynh, N. H. Chu, Y. M. Saputra, D. T. Hoang, D. N. Nguyen, Q.-V. Pham, D. Niyato, E. Dutkiewicz, and W.-J. Hwang, "Transfer Learning for Wireless Networks: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 110, no. 8, pp. 1073–1115, Aug. 2022.

[11] S. Zheng, S. Chen, P. Qi, H. Zhou, and X. Yang, "Spectrum sensing based on deep learning classification for cognitive radios," *IEEE China Communications*, vol. 17, no. 2, pp. 138–148, Feb. 2020.

[12] F. Lin, J. Chen, J. Sun, G. Ding, and L. Yu, "Cross-band spectrum prediction based on deep transfer learning," *IEEE China Communications*, vol. 17, no. 2, pp. 66–80, Feb. 2020.

[13] C. Parera, A. E. Redondi, M. Cesana, Q. Liao, and I. Malanchini, "Transfer Learning for Channel Quality Prediction," in *IEEE International Symposium on Measurements & Networking (M&N 2019)*, Catania, Italy: IEEE, Jul. 2019.

[14] C. Zhang, D. Guo, and P. Fan, "Tracking angles of departure and arrival in a mobile millimeter wave channel," in *IEEE International Conference on Communications (ICC 2016)*, Kuala Lumpur, Malaysia: IEEE, May 2016, pp. 1–6.

[15] V. Va, H. Vikalo, and R. W. Heath, "Beam tracking for mobile millimeter wave communication systems," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP 2016)*, Washington, D.C., United States: IEEE, Dec. 2016, pp. 743–747.

[16] S. Jayaprakasam, X. Ma, J. W. Choi, and S. Kim, "Robust Beam-Tracking for mmWave Mobile Communications," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2654–2657, Dec. 2017.

[17] F. Liu, P. Zhao, and Z. Wang, "EKF-Based Beam Tracking for mmWave MIMO Systems," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2390–2393, Dec. 2019.

[18] J. Lim, H.-M. Park, and D. Hong, "Beam Tracking Under Highly Nonlinear Mobile Millimeter-Wave Channel," *IEEE Communications Letters*, vol. 23, no. 3, pp. 450–453, Mar. 2019.

[19] S. Sun, T. S. Rappaport, R. W. Heath, A. Nix, and S. Rangan, "Mimo for millimeter-wave wireless communications: beamforming, spatial multiplexing, or both?" *IEEE Communications Magazine*, vol. 52, no. 12, pp. 110–121, Dec. 2014.

[20] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11ad Devices," in *13th ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT 2017)*, Incheon, South Korea: ACM, Dec. 2017, pp. 414–425.

[21] IEEE, "802.11ad-2012 - Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band," IEEE, Std 802.11ad-2012, Dec. 2012.

[22] F. Klingler, M. Schettler, S. Dimce, M. S. Amjad, and F. Dressler, "mmWave on the Road – Field Testing IEEE 802.11ad WLAN at 60 GHz," in *40th IEEE International Conference on Computer Communications (INFOCOM 2021), Poster Session*, Virtual Conference: IEEE, May 2021.

[23] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques*, E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez, Eds., Hershey, PA: Information Science Reference, 2009, pp. 242–264.

[24] M. Ranaweera and Q. H. Mahmoud, "Virtual to Real-World Transfer Learning: A Systematic Review," *Electronics*, vol. 10, no. 12, Jun. 2021.