DEEP SEQUENTIAL BEAMFORMER LEARNING FOR MULTIPATH CHANNELS IN MMWAVE COMMUNICATION SYSTEMS

Aditya Sant[†*]*, Afshin Abdi*[†]*, and Joseph Soriaga*[†]

[†]Qualcomm Technologies Inc.
[*]Dept. of Electrical and Computer Engineering, University of California San Diego

ABSTRACT

The highly directional nature of mmWave channels results in a mutlipath incoming signal, often with varying power levels. To exploit the complete diversity of this channel, beamformer design should incorporate this multipath. This increases pilot overhead for initial access. However, low latency mmWave signalling protocols require minimal pilot transmission. Additionally, practical system implementations of beamformers use low-complexity phase-shifter (PS) beamformers. Balancing performance, latency and hardware constraints, active learning has proven to be an extremely effective strategy for initial channel access. However, modern active learning algorithms are tailored to line-of-sight (LoS) angular estimation and tracking. We address multipath beamformer design via deep learning, which is increasingly used for channel estimation and end-to-end communication. We develop two novel deep neural networks (DNN) for multipath beamformer learning: *(i)* Deep Unfolded Beamformer Learning and *(ii)* Deep Recurrent Beamformer Learning. Our approach improves on active learning for LoS paths by utilizing multipath diversity for increased reference signal received power (RSRP).

Index Terms— MmWave Communication, Beamforming, Active Learning, Deep Neural Networks, Recurrent Neural Networks.

1. INTRODUCTION

Modern 5G technologies bring in new advancements for high-speed mmWave communication with a myriad of applications [1–3]. These systems use very directional transmission between both the base station (BS) and the user equipment (UE). The use of massive multiple input multiple output (MIMO) systems enables them to exploit the beamforming gain for robust communication at low SNRs [4, 5].

Due to the physical properties of the mmWave channel [6, 7], consisting of specular multipath components, the initial access procedure and subsequent communication relies on accurate estimation and tracking of the angular components [8–12], while maintaining minimal pilot overhead. Additionally, for reduced power consumption and system complexity, a single RF chain is employed for signal reception via analog beamforming. This makes beamformer design at the receiver a critical stage for the initial access. Active learning for efficient (fixed user-defined) codebook traversal and subsequent angle of arrival estimation balances all the system constraints. One of the most popular methods using active learning for beamforming is the HiePM algorithm [13]. Here, the angular posterior distribution is used as a sufficient statistic for a dyadic codebook traversal. Additional active learning approaches for estimation and tracking have been developed, inspired by this framework [14–17].

This publication was created by Aditya Sant, while an intern at QTI, and currently attends University of California San Diego

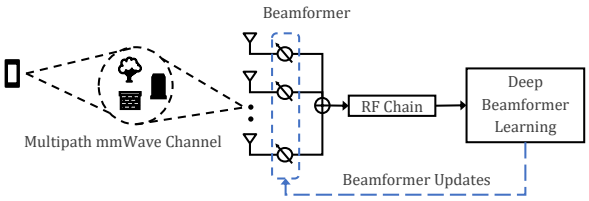

Fig. 1. Multipath uplink mmWave system model

However, most of these algorithms, using a fixed beamforming codebook, cater to static and dyanamic LoS channels. Extending codebook search to accommodate specular multipath components is not scalable, especially if the number of components is unknown. Additionally, utilizing the LoS active learning model for multipath signals will not exploit the complete channel diversity to generate maximal RSRP. For optimal RSRP, we need to match the beamformer to the received channel. We introduce a new framework for multipath beamforming using DNNs, and demonstrate the utility of actively learning a matched filter beamformer. To this end, we develop two independent DNN frameworks: *(i)* Deep Unfolded Beamformer Learning and *(ii)* Deep Recurrent Beamformer Learning, both implemented via PS beamformers, to efficiently solve this problem using few pilot measurements. We avoid the angular decomposition of the multipath received signal to generate the matched filter beamformer. Matched filter beamformer learning thus greatly speeds up the initial access by using fewer pilots. Although the proposed method has been formulated for a general mmWave channel, we numerically test it for a two-path mmWave channel with a fixed power ratio between the components. Through our formulation we lay the groundwork for matched filter beamformer learning, with sufficient scope to extend, generalize and optimize over time.

2. SYSTEM MODEL - TOWARDS MATCHED FILTER BEAMFORMER LEARNING

2.1. System model and beamformer optimization

We consider the multipath wireless sectored channel operating in the EHF $(30-300\,\mathrm{GHz})$ band, as used in [13]. The uplink channel and deep-learning based backend is shown in Fig. 1. The initial access phase consists of T pilot transmissions from the UE, such that the multipath received signal at the BS uniform linear array (ULA) is

$$\mathbf{y}_t = \sum_{i=1}^{L} \sqrt{P_i}\alpha_i \mathbf{a}(\theta_i) + \mathbf{n}_t, \quad t = 0, 1, \ldots, T-1, \qquad (1)$$

where $\mathbf{y}_t$ is the signal received at the ULA at time t, P_i is the power, α_i is the random path gain coefficient and θ_i is the an-

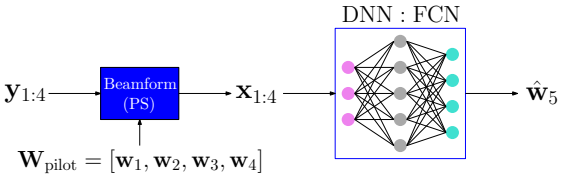

Fig. 2. DNN for matched filter beamforming (shown for $T = 4$)

gle of the i^{th} multipath, L is the number of multipaths, $\mathbf{a}(\theta_i) = \left[1, e^{j\,2\pi d \sin\theta_i/\lambda}, \ldots, e^{j\,2(N-1)\pi d \sin\theta_i/\lambda}\right]^{\mathrm{T}}$ is the receiver ULA manifold with N antennas, and $\mathbf{n}_t \in \mathcal{CN}(\mathbf{0}, \lambda\mathbf{I})$ is the AWGN.

As seen in Fig. 1, the BS performs analog beamforming for the received signal (1). At time instant t, the BS beamformer response is given by $\mathbf{w}_t$, thus the observed signal at the base station is

$$x_t = \mathbf{w}_t^{\mathrm{H}}\mathbf{y}_t = \sum_{i=1}^{L} \sqrt{P_i}\alpha_i \mathbf{w}_t^{\mathrm{H}}\mathbf{a}(\theta_i) + \mathbf{w}_t^{\mathrm{H}}\mathbf{n}_t. \tag{2}$$

To exploit the multipath diversity, the optimal beamformer $\hat{\mathbf{w}}$ is designed to maximize $\text{SNR} = \frac{|\sum_{i=1}^{L}\sqrt{P_i}\alpha_i \mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_i)|^2}{\mathbb{E}_n\,|\mathbf{w}^{\mathrm{H}}\mathbf{n}|^2}$. Since the noise $\mathbf{n}$ is AWGN and the beamformer $\hat{\mathbf{w}}$ is implemented via phase shifters, we can formulate the overall optimization as

$$\begin{aligned} \hat{\mathbf{w}} &= \underset{\mathbf{w}}{\text{argmax}} \left| \sum_{i=1}^{L} \sqrt{P_i}\alpha_i \mathbf{w}^{\mathrm{H}}\mathbf{a}(\theta_i) \right|^2 \\ \text{s.t} \quad & |\mathbf{w}[k]| = \frac{1}{\sqrt{N}}, \quad \forall\, k = 1, 2, \ldots, N. \end{aligned} \tag{3}$$

Matched filter beamformer: Optimal response
The optimal SNR maximizing filter is, by definition, the matched filter [5, 18]; thus the beamformer designed to solve (3) is the matched filter beamformer. Using the notation $\mathbf{y}_s = \sum_{i=1}^{L}\sqrt{P_i}\alpha_i\mathbf{a}(\theta_i)$ for the received signal and P_{s} for the RSRP in (3), we can evaluate the optimal beamformer by simplifying as follows

$$\begin{aligned} P_{\text{s}} = |\mathbf{w}^{\mathrm{H}}\mathbf{y}_s|^2 &= \left| \sum_{k=1}^{N} \mathbf{w}[k]^* \mathbf{y}_s[k] \right|^2 \\ &\le \left[\sum_{k=1}^{N} \left|\mathbf{w}[k]^*\right| . \left|\mathbf{y}_s[k]\right| \right]^2 = \frac{1}{N}\left[\sum_{k=1}^{N} \left|\mathbf{y}_s[k]\right| \right]^2 \end{aligned} \tag{4}$$

where the inequality is via the triangle inequality. Using the inequality (4) and the constraints in (3), the optimal beamformer $\hat{\mathbf{w}}$ is

$$\hat{\mathbf{w}}[n] = \frac{1}{\sqrt{N}} \frac{\mathbf{y}_s[n]}{\left|\mathbf{y}_s[n]\right|}, \quad \forall\, n = 1, 2, \ldots, N. \tag{5}$$

Several methods can extract the matched-filter beamformer from the beamformed observations; chief among them are the compressed sensing based methods [2, 10, 19], that utilize angular sparsity. In our work, we focus on matched filter beamforming via DNNs.

2.2. Matched filter beamforming: DNN-based approach

Unlike active learning for LoS beamformer search (e.g., HiePM [13]), we *design* a matched filter beamformer (see Fig. 2). The received pilots, as in (1), $\mathbf{y}_{1:T} = [\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{T-1}]$ are beamformed using T randomly chosen, PS-based, static pilot beamformers $W_{\text{pilot}} = [\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{T-1}]$, creating the measurements, as per (2), $\mathbf{x} = [x_0, x_1, \ldots, x_{T-1}]^{\mathrm{T}}$, which are fed to the fully connected network (FCN).

The FCN solves the inverse problem, estimating the ideal beamformer $\hat{\mathbf{w}}$, in (5). It is trained on the negative RSRP loss

$$\mathcal{L} = -\frac{1}{N_{\text{train}}} \sum_{m=1}^{N_{\text{train}}} |(\hat{\mathbf{w}}^{(m)})^{\mathrm{H}}\mathbf{y}_s^{(m)}|^2. \tag{6}$$

Specific implementation details are given in Sec. 5. This approach uses a large number of pilots T for near-optimal performance. To minimize pilot usage, we propose two DNN-based sequential learning approaches for the multipath matched filter beamforming, which are further elaborated in the subsequent two sections.

3. DEEP UNFOLDED BEAMFORMER LEARNING

Contrary to the approach in Sec. 2.2, using a fixed static bank of beamformers to estimate the matched filter response, this approach attempts to sequentially build this beamformer bank by utilizing the intermediate estimated beamformers.

Network architecture
The network architecture, for $T = 9$, is shown in Fig. 3(a). The basic building block of the deep unfolded network is the network from Sec. 2.2. Initially, this network, with four static pilot beamformers $\mathbf{W}_{\text{pilot}}$, is used to generate the initial estimate $\hat{\mathbf{w}}_5$ via FCN_1. This is appended to the pilot beamformers as $\hat{\mathbf{W}}_{\text{pilot}} = [\mathbf{W}_{\text{pilot}}, \hat{\mathbf{w}}_5]$, creating a dynamic beamformer bank; the newly received pilot $\mathbf{y}_5$ is beamformed using $\hat{\mathbf{w}}_5$ as $x_5 = \hat{\mathbf{w}}_5^{\mathrm{H}}\mathbf{y}_5$, creating the measurements $\mathbf{x}_{1:5}$. Through FCN_2 $\hat{\mathbf{w}}_6$ is estimated, from input $\mathbf{x}_{1:5}$, and then appended to $\hat{\mathbf{W}}_{\text{pilot}}$. This procedure is repeated for all received pilots till $\mathbf{y}_9$, to generate the final estimated beamformer $\hat{\mathbf{w}}_{10}$.

The network parameters are given in Sec. 5. Through this unfolded network framework, we tailor the intermediate beamformers (and beamformed measurements) to the data, thus significantly reducing the number of pilots used for matched filter beamforming.

Remark 1. This unfolded network, by nature of its design and pretraining strategy, is tailored for a fixed number of pilots. A flexible network design, using variable number of pilots, depending on the SNR, falls outside the scope of this work.

Network training - As opposed to training the entire network end-to-end, we opt for a pretraining strategy for each subnetwork. The sequential pretraining strategy is shown in Algorithm 1. This sequentially builds the complete unfolded network by training each *preceding subnetwork* using the (negative) RSRP loss function (6).
Limitations: Owing to the pretraining and cascading of networks, if a parameter in one of the lower sub-networks is modified, the pretraining for the higher sub-networks has to be repeated. In addition, this is not easily scalable in the *total* number of received pilots T.

To overcome the limitations, we propose a recurrent network to model the estimation as a time series and regress beamformer (5).

4. DEEP RECURRENT BEAMFORMER LEARNING

We now consider the recurrent DNN architecture, that is tailored to preserve estimation history during the forward pass [20–22]. Using this framework we are able to replace the entire cascaded sequence of networks $\{\text{FCN}\}_{k=2}^{6}$ in Fig. 3(a) by a single recurrent network.

Network architecture
The deep recurrent beamformer learning network for $T = 9$ is shown in Fig. 3(b), consisting of two subnetworks

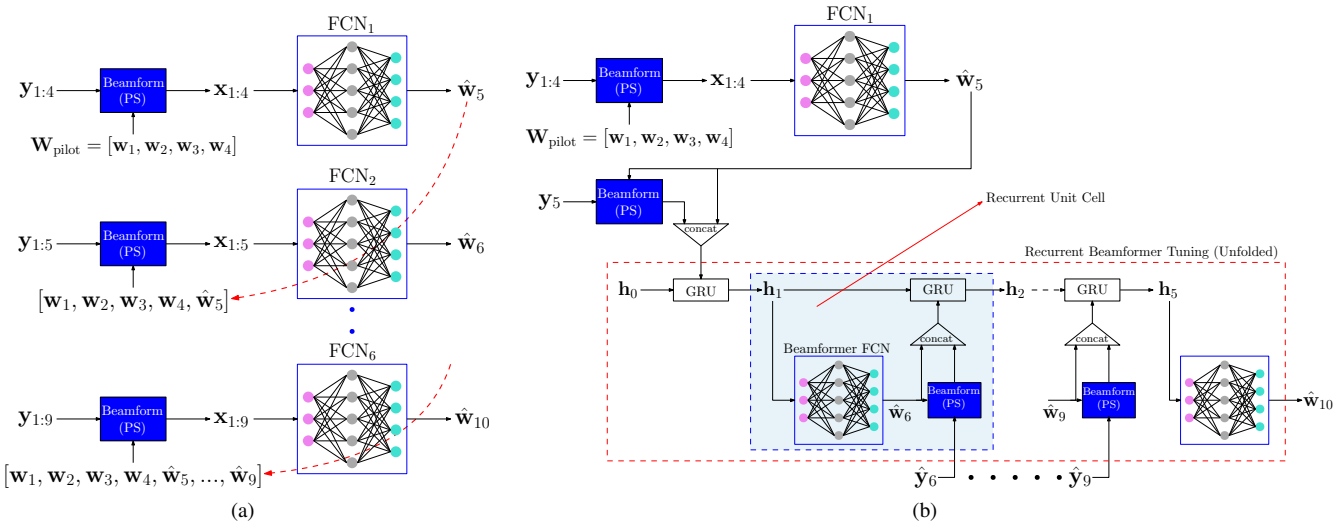

Fig. 3. Deep sequential beamformer learning: (a) Deep Unfolded Beamformer Learning (b) Deep Recurrent Beamformer Learning

Algorithm 1 Pretraining for Deep Unfolded Beamformer Learning

Training data: $\{\mathbf{y}_s^{(m)}\}_{m=1}^{N_{\text{train}}}$, $\{\mathbf{y}_{1:9}^{(m)}\}_{m=1}^{N_{\text{train}}}$
Input: $\mathbf{W}_{\text{pilot}} = [\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4]$, $\{\text{FCN}_i\}_{i=1}^{6}$, N_{epoch}
Output: Net_{out}

for $t = 1$ to N_{epoch} **do** *# static pretraining*
  Train FCN_1 using $\mathcal{L}$ in (6)
end for
$\text{Net}_{\text{out}} \leftarrow \text{FCN}_1$
for $k = 2$ to 6 **do** *# sequential pretraining*
  $\text{Net}_{\text{out}} \leftarrow \text{concat}(\text{Net}_{\text{out}}, \text{FCN}_k)$
  for $t = 1$ to N_{epoch} **do**
    $\hat{\mathbf{w}}^{(m)} \leftarrow \hat{\mathbf{w}}_{k+4}^{(m)}$ *# overall network output*
    Train Net_{out} using $\mathcal{L}$ in (6)
  end for
end for

1. Initialize beamfomer: Initialize beamfomer using a static $\mathbf{W}_{\text{pilot}}$
2. Recurrent beamformer tuning: Using the recurrent unit to iteratively fine tune the initial beamformer estimate

Similar to the deep unfolded net in Fig 3(a), the static beamformer bank W_{pilot} and FCN_1 are used to generate the initial beamformer estimate $\hat{\mathbf{w}}_5$ using the first four pilot symbols $\mathbf{y}_{1:4}$; the network FCN_1 is pretrained using the loss function (6). This initial estimate $\hat{\mathbf{w}}_5$ is now passed to the recurrent network, where it is iteratively refined via a newly received pilot $\{\mathbf{y}_i\}_{i=5}^{9}$ per iteration.

The recurrent unit cell consists of the gated recurrent unit (GRU) and the Beamformer FCN. The input data and hidden state are fed to the GRU, which generates the output hidden state. The sequence of adaptively estimated beamformers, concatenated with the beamformed measurments, i.e., $\{[\hat{\mathbf{w}}_i^{\text{T}}, \hat{\mathbf{w}}_i^H \mathbf{y}_i]^{\text{T}}\}_{i=5}^{9}$, is the time series input to the GRU. The Beamformer FCN transforms the output hidden state $\mathbf{h}_k$ of the GRU to the new beamformer estimate $\mathbf{w}_{k+5}$.

Recurrent unit - GRU: Introduced in [23], this is a highly streamlined recurrent network, offering significantly faster compute times for inference [23, 24]. By virtue of the gating functions, the GRU is able to capture both long and short term memory by resetting and updating the hidden state, using the input. This is potentially useful for the sequential tuning of the matched filter beamformer.

Loss function: We opt for an exponentially weighted loss function

$$\mathcal{L} = \sum_{t=0}^{5} \alpha_t \mathcal{L}_t, \qquad \text{s.t.},$$
$$\mathcal{L}_t = -\frac{1}{N_{\text{train}}} \sum_{m=1}^{N_{\text{train}}} |(\hat{\mathbf{w}}_{t+5}^{(m)})^{\text{H}} \mathbf{y}_s^{(m)}|^2, \tag{7}$$
$$\alpha_t \propto \exp(-\beta(5-t)) \text{ and } \sum_{t=0}^{5} \alpha_t = 1.$$

For this framework, we lack supervised data for the intermediate beamformers $\hat{\mathbf{w}}_5, \ldots, \hat{\mathbf{w}}_9$; we only have the ideal matched filter response (5). Thus, there is no direct way to guide the trajectory for the recurrent inference[1]. By means of the weighted loss function (7), we are able to guide these iterations till the final output $\hat{\mathbf{w}}_{10}$.

5. EXPERIMENTS

5.1. Multipath mmWave channel and experiment parameters

We consider the two-path channel model with received pilots, at the ULA BS with $N = 64$, for $t = 0, 1, \ldots, T-1$ as

$$\mathbf{y}_t = \sqrt{P_1}\alpha_1 \mathbf{a}(\theta_1) + \sqrt{P_2}\alpha_2 \mathbf{a}(\theta_2) + \mathbf{n}_t. \tag{8}$$

The m^{th} training data point $\mathbf{y}_t^{(m)}$ consists of $\alpha_i^{(m)} \in \mathcal{N}(0,1)$, $\theta_i^{(m)} \in \mathcal{U}(-\pi/3, \pi/3)$, independently chosen for each multipath. The signal component for RSRP evaluation for each measurement is $\mathbf{y}_s^{(m)} = \sqrt{P_1}\alpha_1^{(m)}\mathbf{a}(\theta_1^{(m)}) + \sqrt{P_2}\alpha_2^{(m)}\mathbf{a}(\theta_2^{(m)})$. The number of pilots T is varied for different experiments described in this section. As stated in Sec. 2.1, we consider $P_1 = 1$ and $P_1/P_2 = 2$ as constant for training and testing. The noise variance λ for $\mathbf{n}_t$ is varied such that the input $\text{SNR} = \frac{\mathbb{E}\,||\mathbf{y}_s||^2}{\mathbb{E}\,||\mathbf{n}_t||^2}$ varies from -10 to 15 dB.

Active learning baseline - HiePM [13]: For the received signal (8), we use HiePM, without any changes, to recover the component with the larger power, i.e., $\sqrt{P_1}\alpha_1 \mathbf{a}(\theta_1)$. We run the HiePM algorithm

[1]For the deep unfolded beamformer learning net, this is avoided by the use of the pretraining (Algo 1).

Table 1. DNN Architecture Summary

Network	Sub-network	Parameters
1) Static Beamformer Learning	-	Input neurons: 8
		Hidden layers: 3
		Hidden neurons: 512
		Hidden layers: FCN + ReLU + BN
		Output layer normalize: $\lvert\hat{\mathbf{w}}[i]\rvert = \frac{1}{\sqrt{N}}$
2) Deep Unfolded Beamformer Learning (K subnetworks)	$\{\text{FCN}_k\}_{k=1}^{6}$	Input neurons: $8 + 2 * (k-1)$
		Hidden state and output same as 1)
		Output normalize same as 1)
3) Deep Recurrent Beamformer Learning (K iterations)	FCN_1	Same as FCN_1 of 2)
	Beamformer FCN	Input neurons: 1024
		Hidden layers: 2
		Hidden neurons: 512
		Hidden layer: FCN + ReLU + BN
		Output normalize same as 1)
	GRU	Hidden state dimension: 1024
		Input dimension: 130
		β in (7) = 3

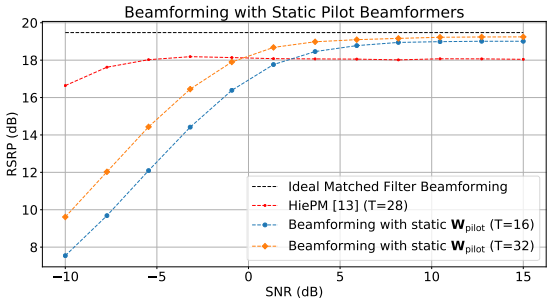

Fig. 4. Matched filter beamformer learning using static $\mathbf{W}_{\text{pilot}}$

for a fixed time frame of $T = 28$. The same dyadic codebook as in [13] is used with a total 7 layers; thus the final layer of the codebook partitions the sector $[-\pi/3, \pi/3]$ into 128 segments. As per the implementation in [13], the HiePM algorithm has knowledge of P_1, α_1 and λ, resulting in improved performance of HiePM at lower SNRs as compared matched filter beamformer learning. However, this is not practical for multipath channel access. For HiePM, we also omit the implementation of the beamformers via PS hardware.

DNN parameters: The parameters for all the networks are summarized in Table 1. Although the inputs and outputs of the DNNs are complex-valued, we consider real-valued weights for the networks. Thus we split the complex input (and output) into the constituent real and imaginary parts as: $\hat{\mathbf{x}} = [\mathfrak{Re}(\mathbf{x})^{\mathrm{T}}, \mathfrak{Im}(\mathbf{x})^{\mathrm{T}}]^{\mathrm{T}}$. All the networks are trained in the SNR range $5 - 10\,\text{dB}$ and tested in $-10 - 15\,\text{dB}$.

5.2. Simulation results

1) Matched filter beamforming for multipath (static $\mathbf{W}_{\text{pilot}}$): The plots in Fig. 4 illustrate the utility of matched filter beamforming for multipath channels when compared to the HiePM implementation described above. The ideal matched-filtering plot is the upper bound on the RSRP, given by (4), using the ideal beamformer (5). Plots for the matched filter beamforming using static $\mathbf{W}_{\text{pilot}}$ (see Sec. 2.2) show that it is possible to *learn* this ideal beamformer. While the HiePM algorithm is limited by the dominant path, match filter beamforming exploits the multipath diversity, increasing the RSRP. However, learning the matched filter beamformer using a static $\mathbf{W}_{\text{pilot}}$ needs a large number of pilots for near-optimal performance; we now illustrate sequential learning to further reduce the pilots required.

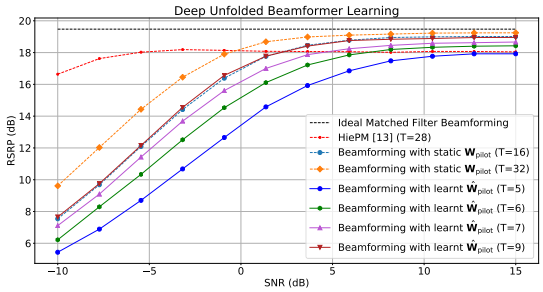

Fig. 5. Performance for deep unfolded beamformer learning

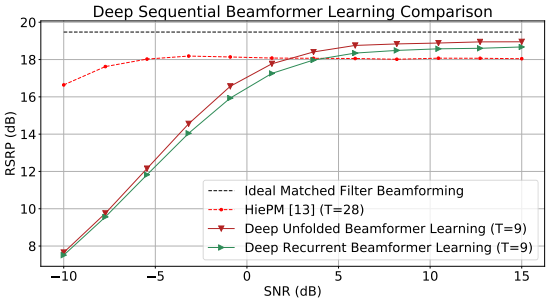

Fig. 6. Comparing performance for deep sequential learning

2) Sequential beamformer learning (dynamic $\hat{\mathbf{W}}_{\text{pilot}}$): The plots in Fig. 5 analyze the performance of the deep unfolded beamforming network (see Sec 3) for different number of pilots T. Compared to the HiePM algorithm, active learning with just 7 pilots is able to exploit the multipath diversity and increase RSRP at moderate SNR. Additionally, compared to matched filter beamformer learning with static $\mathbf{W}_{\text{pilot}}$, the sequential strategy reaches near-optimal performance with significantly fewer pilots. Thus the unfolded network, by tailoring $\hat{\mathbf{W}}_{\text{pilot}}$ to the data, can reduce pilot usage by over 40%.

Finally, the performance plots in Fig. 6 illustrate that the time series modeling through deep recurrent networks can improve on the complexity of unfolded learning architecture in Fig. 3(a), without significantly sacrificing on RSRP. Using the flexibility of the deep recurrent beamformer network, we will be able to cater to variable number of pilots, proving invaluable for practical implementations.

6. CONCLUSION

Through this work we extend the conventional LoS active learning to multipath match filter beamforming, practically implemented via analog PS-based hardware, via DNNs. Although we are able to learn the ideal matched filter beamformer response through a static pilot beamformer bank, this utilizes a large number of pilot symbols. To overcome this problem we introduced a sequential learning strategy via a deep unfolded network and a deep recurrent network. Through our results we are able to show more than 40% reduction in pilot usage through this sequential learning strategy. Further, by means of the deep recurrent framework, we introduce a flexible architecture for initial access. Future directions for this work involve further optimizing these networks in order to cater to general multipath channels as well as variable time frames for initial access.

7. REFERENCES

[1] Sundeep Rangan, Theodore S Rappaport, and Elza Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[2] Robert W Heath, Nuria Gonzalez-Prelcic, Sundeep Rangan, Wonil Roh, and Akbar M Sayeed, "An overview of signal processing techniques for millimeter wave mimo systems," *IEEE journal of selected topics in signal processing*, vol. 10, no. 3, pp. 436–453, 2016.

[3] Theodore S Rappaport, Yunchou Xing, George R MacCartney, Andreas F Molisch, Evangelos Mellios, and Jianhua Zhang, "Overview of millimeter wave communications for fifth-generation (5g) wireless networks—with a focus on propagation models," *IEEE Transactions on antennas and propagation*, vol. 65, no. 12, pp. 6213–6230, 2017.

[4] Harry L Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*, John Wiley & Sons, 2004.

[5] David Tse and Pramod Viswanath, *Fundamentals of wireless communication*, Cambridge university press, 2005.

[6] G Allen and A Hammoudeh, "Outdoor narrow band characterisation of millimetre wave mobile radio signals," in *IEE Colloquium on Radiocommunications in the Range 30-60 GHz*. IET, 1991, pp. 4–1.

[7] Sooyoung Hur, Sangkyu Baek, Byungchul Kim, Youngbin Chang, Andreas F Molisch, Theodore S Rappaport, Katsuyuki Haneda, and Jeongho Park, "Proposal on millimeter-wave channel modeling for 5g cellular system," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 454–469, 2016.

[8] Xiaoshen Song, Saeid Haghighatshoar, and Giuseppe Caire, "Efficient beam alignment for millimeter wave single-carrier systems with hybrid mimo transceivers," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1518–1533, 2019.

[9] Ahmed Alkhateeb, Omar El Ayach, Geert Leus, and Robert W Heath, "Channel estimation and hybrid precoding for millimeter wave cellular systems," *IEEE journal of selected topics in signal processing*, vol. 8, no. 5, pp. 831–846, 2014.

[10] Yacong Ding and Bhaskar D Rao, "Channel estimation using joint dictionary learning in fdd massive mimo systems," in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2015, pp. 185–189.

[11] Chun-Lin Liu and PP Vaidyanathan, "One-bit sparse array doa estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3126–3130.

[12] Aditya Sant and Bhaskar D Rao, "Doa estimation in systems with nonlinearities for mmwave communications," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 4537–4541.

[13] Sung-En Chiu, Nancy Ronquillo, and Tara Javidi, "Active learning and csi acquisition for mmwave initial alignment," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2474–2489, 2019.

[14] Nancy Ronquillo and Tara Javidi, "Active beam tracking under stochastic mobility," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–7.

[15] Jianjun Zhang, Yongming Huang, Yu Zhou, and Xiaohu You, "Beam alignment and tracking for millimeter wave communications via bandit learning," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5519–5533, 2020.

[16] Nancy Ronquillo, Sung-En Chiu, and Tara Javidi, "Sequential learning of csi for mmwave initial alignment," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1278–1283.

[17] Foad Sohrabi, Zhilin Chen, and Wei Yu, "Deep active learning approach to adaptive beamforming for mmwave initial alignment," *IEEE Journal on Selected Areas in Communications*, 2021.

[18] George Turin, "An introduction to matched filters," *IRE transactions on Information theory*, vol. 6, no. 3, pp. 311–329, 1960.

[19] Mark A Davenport, Michael B Wakin, and Richard G Baraniuk, "The compressive matched filter," *Rice Univ., Houston, USA, Tech. Rep. TREE*, vol. 610, 2006.

[20] Ahmed Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, 2018.

[21] Zachary C Lipton, John Berkowitz, and Charles Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019*, 2015.

[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.

[23] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[24] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.