Active Sensing for Communications by Learning

Wei Yu

Joint work with Foad Sohrabi, Zhilin Chen, Tao Jiang, Wei Cui

University of Toronto

January 2022

Integrated Communications and Sensing

- Why integrated communications and sensing?
  - Integrated mmWave/THz spectrum
  - Integrated transceiver hardware
  - Integrated waveform
  - Integrated applications
- Most communication systems already have both sensing and communication capabilities:
  - Sensing: Channel estimation
  - Communication: Data transmission
- Separation principles have always guided classical communications system design:
  - Separation of source coding and channel coding
  - Separation of channel estimation and data transmission
  - Separation of coding and modulation
- This talk is about sensing for communications.

Role of Machine Learning in Optimizing Sensing Strategies

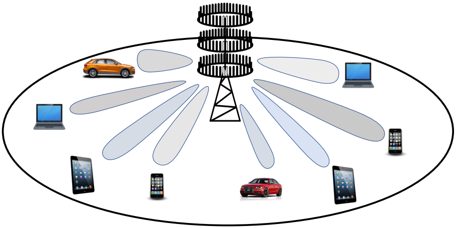
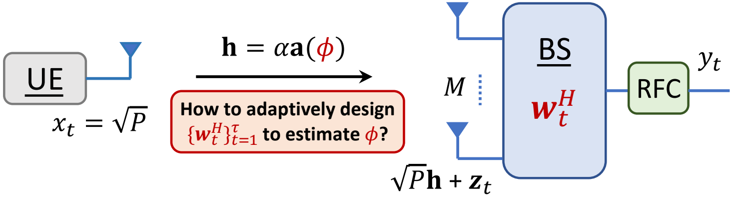


Fig.: Cellular base-station with a large-scale antenna array.

- Motivation: mmWave massive MIMO for enhanced mobile broadband.
- **Estimating high-dimensional channel from low-dimensional observations is challenging:**
  - Fully digital beamforming: Requires one high-resolution RF chain per antenna element.
  - Hybrid beamforming: Analog beamformer with low-dimensional digital beamforming.
- **Initial Beam Alignment:** How to find channel direction in a system with limited RF chains?

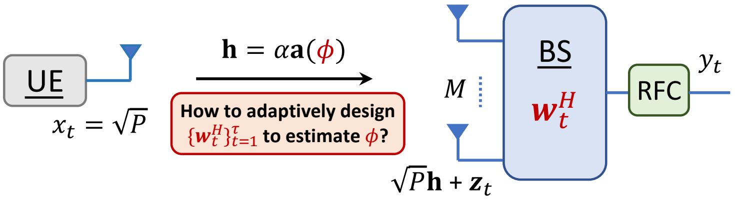
Sensing Architecture with Hybrid Beamforming



- A BS with M antennas and a single RF chain serves a single-antenna user
- The user transmits pilot; the BS tries to estimate the channel.
- Due to the RF chain limitation, the BS must sense the channel through analog combiners:

$$y_t = \mathbf{w}_t^H \mathbf{h} x_t + \mathbf{w}_t^H \mathbf{z}_t = \sqrt{P}\alpha\ \mathbf{w}_t^H a(\phi) + n_t, \tag{1}$$

  - $\mathbf{w}_t$ is the sensing (combining) vector in time frame t with $\|\mathbf{w}_t\|^2 = 1$
  - $\alpha \sim \mathcal{CN}(0, 1)$ is the fading coefficient,
  - $\phi \in [\phi_{\text{min}}, \phi_{\text{max}}]$ is the angle of arrival (AoA),
  - $\mathbf{a}(\phi) = \left[1, e^{j\pi \sin\phi}, \ldots, e^{j(M-1)\pi \sin\phi}\right]^T$ is the array response vector,
  - $n_t \sim \mathcal{CN}(0, 1)$ is the effective white Gaussian noise.



- **Initial Beam Alignment:** The BS can optimize the quality of AoA estimation by designing $\mathbf{w}_t$ at each time frame, possibly sequentially in an adaptive manner, i.e.,

$$\mathbf{w}_{t+1} = \widetilde{\mathcal{G}}_t\left(y_{1:t}, \mathbf{w}_{1:t}\right), \quad \forall t \in \{0, \ldots, \tau - 1\}. \tag{2}$$

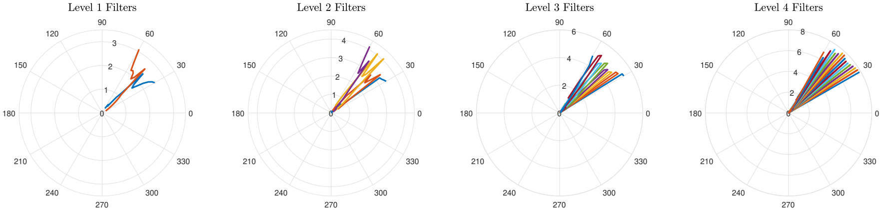
- The final AoA estimate is obtained as a function of all past observations as:

$$\hat{\phi} = \widetilde{\mathcal{F}}\left(y_{1:\tau}, \mathbf{w}_{1:\tau}\right). \tag{3}$$

- This is a high-dimensional sequential decision problem!

Traditional Approach: Bisection in Angle Domain

- We can select the sensing vector from a pre-designed codebook that minimizes the expected MSE objective, e.g., the codebook contains the following 30 filters bisecting in angle domain.



- Hierarchical beamforming codebook [Alkhateeb, Ayach, Leus, and Heath, 2014].

Is this hierarchical approach optimal?

- **Goal:** To adaptively and sequentially design sensing vectors $\mathbf{w}_t$ to optimize AoA estimation.
- **Observation:** Since the AoA posterior distribution $\boldsymbol{\pi}_t$ provides sufficient statistic for adaptive sensing in initial alignment [Chiu, Ronquillo, and Javidi, 2019], we alternatively consider:

$$\mathbf{w}_{t+1} = \mathcal{G}_t\left(\boldsymbol{\pi}_t\right), \qquad \hat{\phi} = \mathcal{F}\left(\boldsymbol{\pi}_\tau\right), \tag{4}$$

where $\mathcal{G}_t(\cdot)$ is the adaptive sensing strategy and $\mathcal{F}(\cdot)$ is the AoA estimation scheme.

- **Problem Formulation:** MMSE estimation of AoA

$$\min_{\{\mathcal{G}_t(\cdot)\}_{t=0}^{\tau-1}, \mathcal{F}(\cdot)} \quad \mathbb{E}\left[\left(\hat{\phi} - \phi\right)^2\right] \tag{5a}$$

$$\text{s.t.} \quad \mathbf{w}_{t+1} = \mathcal{G}_t\left(\boldsymbol{\pi}_t\right), \quad \forall t \in \{0, \ldots, \tau-1\}, \tag{5b}$$

$$\hat{\phi} = \mathcal{F}\left(\boldsymbol{\pi}_\tau\right). \tag{5c}$$

- The joint design of the adaptive beamforming strategy $\{\mathcal{G}_t(\cdot)\}_{t=0}^{\tau-1}$ and the AoA estimation scheme $\mathcal{F}(\cdot)$ by directly solving (5) is challenging.
- Codebook-based beamforming for initial beam alignment is NOT optimal.
  - Much more efficient codebook-free adaptive beamforming scheme can be designed.
  - A deep learning framework can be used to efficiently design adaptive sensing strategy.

Motivating Example: Beamforming Design in One Stage

- To see the benefits of utilizing the deep learning framework for the beam alignment problem, let's formulate the problem for the last time frame. **(For simplicity, assume $\alpha = 1$.)**
- Given a prior distribution $\boldsymbol{\pi}_{\tau-1}(\phi)$, how to design $\mathbf{w}_\tau$ to minimize AoA MSE at next stage τ?
- Given the sensing vector $\mathbf{w}_\tau$, the posterior distribution given the received signal y_τ is:

$$\boldsymbol{\pi}_\tau(\phi|y_\tau) = \frac{\boldsymbol{\pi}_{\tau-1}(\phi) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\phi)\|^2}}{\int_{\phi_{\min}}^{\phi_{\max}} \boldsymbol{\pi}_{\tau-1}(\tilde{\phi}) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\tilde{\phi})\|^2} d\tilde{\phi}}. \tag{6}$$

- Now, the best MSE estimate for ϕ is given by:

$$\hat{\phi} = \mathbb{E}[\phi|y_\tau] = \frac{\int_{\phi_{\min}}^{\phi_{\max}} \phi \ \boldsymbol{\pi}_{\tau-1}(\phi) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\phi)\|^2} d\phi}{\int_{\phi_{\min}}^{\phi_{\max}} \boldsymbol{\pi}_{\tau-1}(\tilde{\phi}) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\tilde{\phi})\|^2} d\tilde{\phi}}. \tag{7}$$

- Therefore, the one-stage beam alignment problem can be written as:

$$\min_{\mathbf{w}_\tau} \int_{\phi_0} \int_y \left(\frac{\int \phi \ \boldsymbol{\pi}_{\tau-1}(\phi) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\phi)\|^2} d\phi}{\int \boldsymbol{\pi}_{\tau-1}(\tilde{\phi}) e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\tilde{\phi})\|^2} d\tilde{\phi}} - \phi_0 \right)^2 \cdot \underbrace{\frac{1}{\pi} e^{-\|y_\tau - \sqrt{P}\mathbf{w}_\tau^H \mathbf{a}(\phi_0)\|^2} \boldsymbol{\pi}_{\tau-1}(\phi_0)}_{\text{joint distribution of } y_\tau \text{ and } \phi_0} dy_\tau \, d\phi_0$$

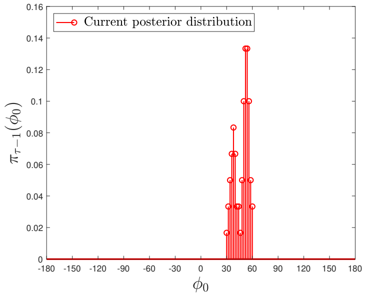
$$\text{s.t.} \quad \|\mathbf{w}_\tau\|^2 = 1.$$

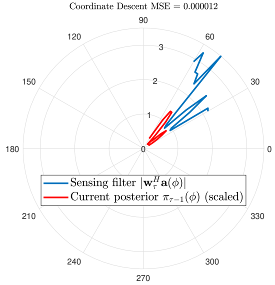
- The sensing design problem for beam alignment, even for a single time frame, is complicated!
- **Question:** How to solve this problem?

$$\min_{\mathbf{w}_\tau} \int_{\phi_0}\int_{y}\left(\frac{\int \phi\ \boldsymbol{\pi}_{\tau-1}(\phi)e^{-\|y_\tau-\sqrt{P}\mathbf{w}_\tau^H\mathbf{a}(\phi)\|^2}d\phi}{\int \boldsymbol{\pi}_{\tau-1}(\tilde{\phi})e^{-\|y_\tau-\sqrt{P}\mathbf{w}_\tau^H\mathbf{a}(\tilde{\phi})\|^2}d\tilde{\phi}}-\phi_0\right)^2\cdot\frac{1}{\pi}e^{-\|y_\tau-\sqrt{P}\mathbf{w}_\tau^H\mathbf{a}(\phi_0)\|^2}\boldsymbol{\pi}_{\tau-1}(\phi_0)dy_\tau d\phi_0$$

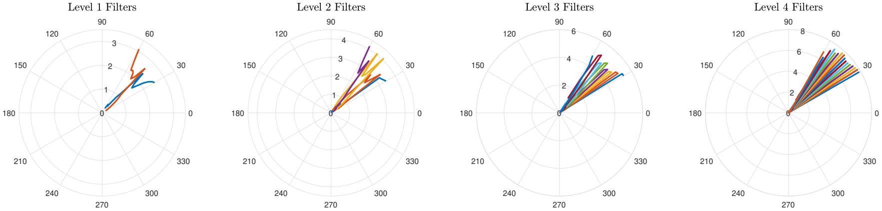
$$\text{s.t.}\quad \|\mathbf{w}_\tau\|^2=1.$$

- We can perform a coordinate descent (CD) algorithm to design $\mathbf{w}_\tau$.
- Example: We consider a simple prior distribution as in the below figure and set $P=10$.
- It takes few days for the CD algorithm (on CPU) to converge to the solution!

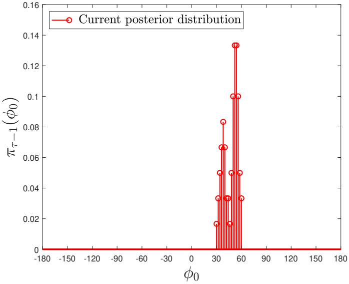


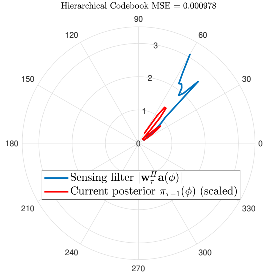


- We can select the sensing vector from a pre-designed hierarchical codebook, which minimizes the expected MSE objective, i.e., we can select one of the following 30 filters.

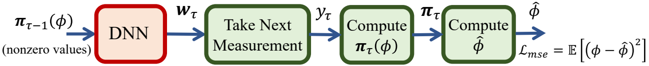


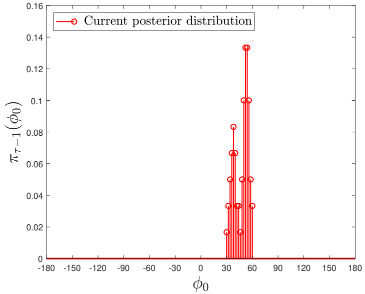
- Although utilizing the hierarchical codebook significantly reduces the computational complexity, the achieved MSE is much worse than the CD algorithm (by 18.5dB).

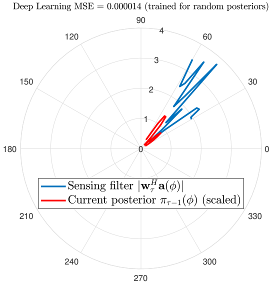


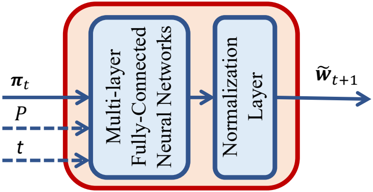


- We can use a deep neural network to design $\mathbf{w}_\tau$.
- Generalizability: The DNN can be trained for a class of $\pi_{\tau-1}(\phi)$ (instead of a particular one).
- Efficiency: It takes a few minutes to train a DNN to converge to the same MSE as CD!





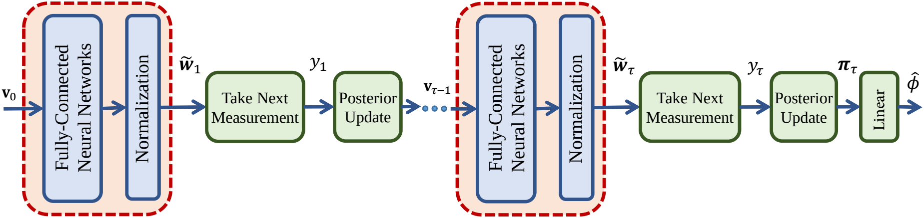




- We propose a DNN with L dense layers to undertake the adaptive sensing design.
- **DNN Inputs:** The current posterior distribution together with the other available system parameters, i.e., $\mathbf{v}_t = \left[\boldsymbol{\pi}_t^T, P, t\right]^T$.
- **DNN Output:** The beamforming vector for the next measurement:

$$\widetilde{\mathbf{w}}_{t+1} = \sigma_L\left(\mathbf{A}_L \sigma_{L-1}\left(\cdots \sigma_1\left(\mathbf{A}_1 \mathbf{v}_t + \mathbf{b}_1\right)\cdots\right) + \mathbf{b}_L\right), \tag{10}$$

  - $\sigma_\ell, \mathbf{A}_\ell$, and $\mathbf{b}_\ell$ are the activation function, weights, and biases in the ℓ-th layer.
  - $\widetilde{\mathbf{w}}_{t+1}$ is the real representation of the next beamforming vector.
  - Normalization layer, $\sigma_L(\cdot) = \frac{\cdot}{\|\cdot\|}$, ensures that the power constraint is met.
- **Question:** How to train this DNN?



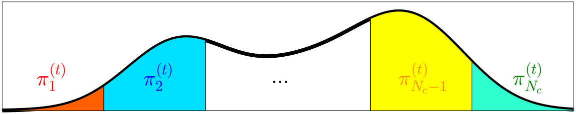
- By considering all τ beamforming stages, we can think of the proposed end-to-end architecture as a very deep neural network architecture.
- The last linear layer outputs the AoA estimate based on the posterior distribution.
- The ultimate goal of this DNN is to successfully recover the AoA value. Thus, we use:

$$\mathcal{L} = -\mathbb{E}\left[(\phi - \hat{\phi})^2\right]. \tag{11}$$

- The DNN is trained using stochastic gradient descent (SGD) to minimize the above loss.

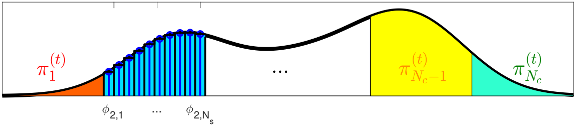
Discretizing the Posterior Distribution of AoA

- The prior and posterior distributions are probability density functions (PDF).
- The prior distribution needs to be discretized before being passed to the DNN:



- To compute the posterior after DNN, we further approximate by using the midpoint rule:

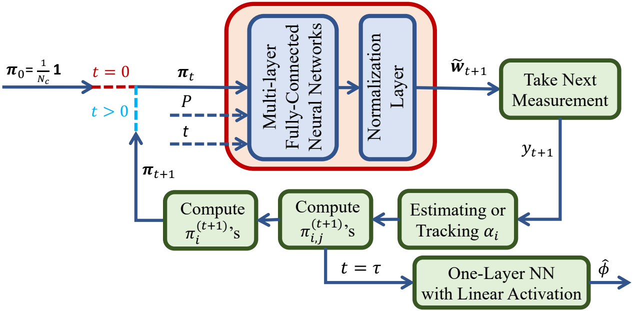
$$\pi_i^{(t)} \approx \sum_{j=1}^{N_s} \pi_{i,j}^{(t)}, \quad \text{where} \quad \pi_{i,j}^{(t)} = \frac{\sum_{j=1}^{N_s} \pi_{i,j}^{(t-1)} e^{-\|y_t - \sqrt{P}\alpha \mathbf{w}_t^H \mathbf{a}(\phi_{i,j})\|^2}}{\sum_{i=1}^{N_c} \sum_{j=1}^{N_s} \pi_{i,j}^{(t-1)} e^{-\|y_t - \sqrt{P}\alpha \mathbf{w}_t^H \mathbf{a}(\phi_{i,j})\|^2}}. \tag{12}$$



- The choices of the number of AoA intervals, N_c, and the number samples within each interval, N_s, provide an accuracy-complexity trade-off.

Fading Coefficient Estimation

- Fading coefficient α is also unknown. Computing the posterior exactly involves integration.



- To address the computational complexity issue,
  1. The fading coefficient is estimated in each time frame.
  2. The estimate of the fading coefficient is then used to compute the AoA posterior.
- We investigate two different estimation strategies:
  1. MMSE estimator,
  2. Kalman filter.

MMSE Estimation of Fading Coefficient

- Assuming that the true AoA is ϕ_i, we seek to estimate the fading coefficient at time frame t from the available measurements, i.e.,

$$\mathbf{y}_t = \mathbf{c}_{i,t}\alpha + \mathbf{n}_t, \tag{13}$$

where $\mathbf{y}_t = [y_1, \ldots, y_t]^T$, $\mathbf{c}_{i,t} = \sqrt{P}\mathbf{W}_t^H \mathbf{a}(\phi_i)$ with $\mathbf{W}_t \triangleq [\mathbf{w}_1, \ldots, \mathbf{w}_t]$, and $\mathbf{n}_t \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$.

- Given $\alpha \sim \mathcal{CN}(0, 1)$, the best MSE estimate of the fading coefficient can be computed as:

$$\hat{\alpha}_i^{(t)} = \left(\mathbf{c}_{i,t}^H \mathbf{c}_{i,t} + 1\right)^{-1} \mathbf{c}_{i,t}^H \mathbf{y}_t. \tag{14}$$

- This MMSE estimate can then be used to approximate the AoA posterior distribution as:

$$\pi_i^{(t+1)} = \frac{\prod_{\tilde{t}=1}^{t+1} e^{-\|y_{\tilde{t}} - \sqrt{P}\hat{\alpha}_i^{(t+1)} \mathbf{w}_{\tilde{t}}^H \mathbf{a}(\phi_i)\|^2}}{\sum_{j=1}^{N} \prod_{\tilde{t}=1}^{t+1} e^{-\|y_{\tilde{t}} - \sqrt{P}\hat{\alpha}_j^{(t+1)} \mathbf{w}_{\tilde{t}}^H \mathbf{a}(\phi_j)\|^2}}. \tag{15}$$

- We numerically observe that this MMSE approach leads to an excellent performance.
- However, this approach requires:
  - High memory usage to store all the received signals $\mathbf{y}_t$ and sensing vectors $\mathbf{W}_t$.
  - High computational complexity to compute (14) and (15), i.e., $O(\tau^2 MN)$.
- **Question:** Alternative estimation method with lower storage and computational complexity?

Fading Coefficient Estimation via Kalman Filter

- We first assume that the conditional probability density of the fading coefficient given ϕ_i and $\mathbf{y}_t$ is complex Gaussian with mean $\mu_{\alpha,i}^{(t)}$ and variance $\gamma_{\alpha,i}^{(t)}$.
- We then exploit the concept of Kalman filtering to estimate those means and variances as:

$$\mu_{\alpha,i}^{(t+1)} = \mu_{\alpha,i}^{(t)} + \frac{\gamma_{\alpha,i}^{(t)} g_{i,t+1}^{*}}{\gamma_{\alpha,i}^{(t)} |g_{i,t+1}|^2 + 1} (y_{t+1} - \mu_{\alpha,i}^{(t)} g_{i,t+1}), \tag{16a}$$

$$\gamma_{\alpha,i}^{(t+1)} = \gamma_{\alpha,i}^{(t)} \frac{1}{\gamma_{\alpha,i}^{(t)} |g_{i,t+1}|^2 + 1}, \tag{16b}$$

where $g_{i,t+1} = \sqrt{P}\ \mathbf{w}_{t+1}^{H} \mathbf{a}(\phi_i)$, $\mu_{\alpha,i}^{(0)} = 0$, and $\gamma_{\alpha,i}^{(0)} = 1, \forall i$.

- With the above assumptions in place, the AoA posterior update rule can be written as:

$$\pi_i^{(t+1)} = \frac{\pi_i^{(t)}\ e^{\frac{-\|y_{t+1} - \mu_{\alpha,i}^{(t+1)} g_{i,t}\|^2}{\gamma_{\alpha,i}^{(t+1)} |g_{i,t+1}|^2 + 1}}}{\sum_{j=1}^{N} \pi_j^{(t)} e^{\frac{-\|y_{t+1} - \mu_{\alpha,j}^{(t+1)} g_{j,t}\|^2}{\gamma_{\alpha,i}^{(t+1)} |g_{j,t+1}|^2 + 1}}}, \tag{17}$$

- Advantages of the Kalman filter approach:
  - Posterior distribution is updated sequentially without having to store past $\mathbf{W}_t$'s and $\mathbf{y}_t$'s.
  - The overall computational complexity for posterior computation is reduced to $O(\tau MN)$.

Implementation Details

- **Implementation platform:** TensorFlow and Keras.
- **Optimization method:** Adam optimizer with an adaptive learning rate initialized to 0.001.
- $\#$ **hidden layers:** $L = 4$.
- $\#$ **hidden neurons/layer:** $[1024, 1024, 1024, 2M]$.
- **Input size:** $N + 2$.
- **Activation function of the hidden layers:** Rectified linear units (ReLUs).
- In the training stage, the transmit power is generated so that:

$$\text{SNR} \triangleq 10 \log_{10}(P) \in \mathcal{U}(-10, 25) dB.$$

- The single-path channel parameters are set to:
  - $N = 128$,
  - $\alpha \sim \mathcal{CN}(0, 1)$,
  - $\phi_{\text{min}} = -60^\circ$ and $\phi_{\text{max}} = 60^\circ$.

Numerical Results

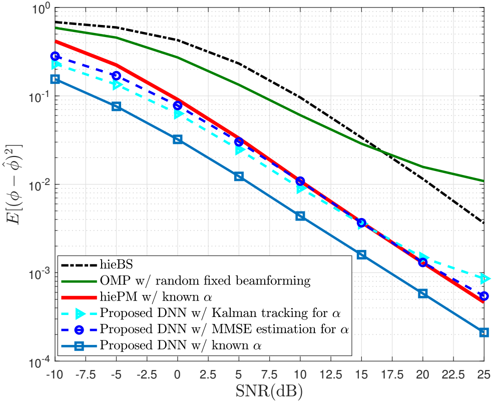
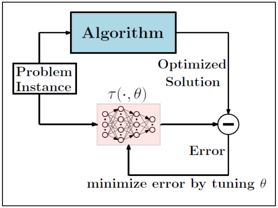


Fig.: MSE versus SNR in a system with $M = 64$ and $\tau = 14$. Here, we set $N_c = 128$ and $N_s = 20$.

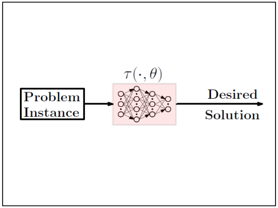
- **Baselines:**
  - Compressive sensing with fixed beamforming, e.g., OMP [Tropp, Gilbert, 2007].
  - Hierarchical codebook with bisection search (hieBS) [Alkhateeb, Ayach, Leus, Heath, 2014].
  - Hierarchical codebook with posterior matching (hiePM) [Chiu, Ronquillo, Javidi, 2019].

Role of Machine Learning in Sensing and Communications

- Traditional optimization is about efficient search through an optimization landscape
  - The holy grail of optimization is to transform a problem into convex form.
  - There is no universal theory about how to best transform the optimization landscape.
- Machine learning enables data-driven optimization
  - Complexity is moved from the optimization step to the neural network training process.
  - Once trained, neural network directly maps problem parameters to optimized solution.
  - The task of optimization is turned into "pattern matching".
  - Neural network is a universal model with a large number of trainable parameters.

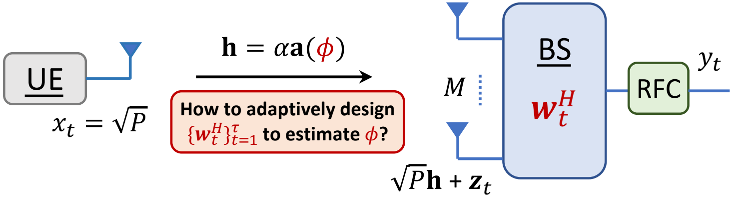


(a) Training Stage



(b) Testing Stage

Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, Nicholas D. Sidiropoulos, "Learning to Optimize: Training Deep Neural Networks for Wireless Resource Management", IEEE Transactions on Signal Processing, vol. 66, no. 20, pp. 5438-5453, October 15, 2018. (Figure credit)

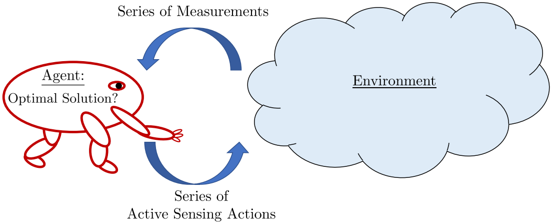


- **Initial Beam Alignment:** BS sequentially design $\mathbf{w}_t$ to minimize MSE of estimating ϕ:

$$\mathbf{w}_{t+1} = \widetilde{\mathcal{G}}_t \left(y_{1:t}, \mathbf{w}_{1:t} \right), \quad \forall t \in \{0, \dots, \tau - 1\}.$$

- The proposed machine learning approach already outperforms the state-of-the-art.
- But the complexity of computing posterior distribution is high.
  - Posterior distribution is impossible to compute when there are multiple paths.
- Can the computation of the posterior distribution be avoided?
- Can performance be improved by matching neural network architecture to problem structure?

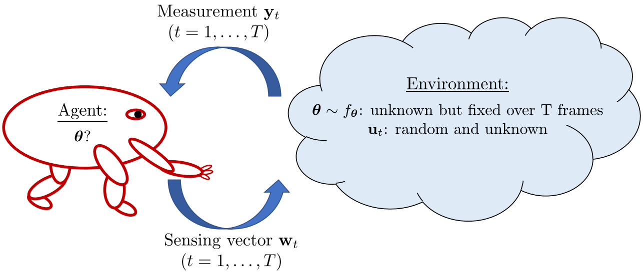
Active Sensing Problem



- The active sequential learning problem naturally arises in many inference, sensing, and control settings, e.g., tree-search, sequential design of experiments, the multi-armed bandit.
- Problems involve adaptive estimation/control based on sequential sensing of environment.
- Analytic solutions are in general not available.
- Numerical solutions are computationally complex and in general hard to obtain.

What is the best machine learning model for finding the optimal sequential sensing actions efficiently?

Active Sensing Problem Formulation



- We consider an active sensing setup where an agent interacts with an environment over T time frames to estimate a system parameter $\boldsymbol{\theta} \in \mathbb{R}^N$.
- In each time frame t, the agent designs a sensing vector $\mathbf{w}_t \in \mathbb{R}^M$ and subsequently observes a measurement $\mathbf{y}_t \in \mathbb{R}^D$ as:

$$\mathbf{y}_t = \mathcal{H}\left(\mathbf{w}_t, \boldsymbol{\theta}, \mathbf{u}_t\right), \quad t = 1, \dots, T. \tag{18}$$

  where $\mathbf{u}_t \in \mathbb{R}^U$ is the additional stochastic parameters of the system.
- The agent can sequentially design $\mathbf{w}_t$ at each time frame, possibly in an adaptive manner:

$$\mathbf{w}_{t+1} = \mathcal{G}_t\left(\mathbf{y}_{1:t}, \mathbf{w}_{1:t}\right), \quad t = 0, \dots, T-1. \tag{19}$$

- The final estimate of $\boldsymbol{\theta}$ is obtained as a function of all past observations:

$$\hat{\boldsymbol{\theta}} = \mathcal{F}\left(\mathbf{y}_{1:T}, \mathbf{w}_{1:T}\right). \tag{20}$$

General Objective Function

- In many applications, instead of directly estimating $\boldsymbol{\theta}$, the agent may be interested in designing a control action $\mathbf{v} \in \mathbb{R}^V$ in order to maximize a utility function $\mathcal{J}(\boldsymbol{\theta}, \mathbf{v})$.
- For example, in the mmWave initial alignment problem, the control action can be the beamforming vector for subsequent data transmission.
- For these settings, the problem of interest can be formulated as:

$$\begin{aligned}
&\underset{\{\mathcal{G}_t(\cdot,\cdot)\}_{t=0}^{T-1}, \mathcal{F}(\cdot,\cdot)}{\text{maximize}} \quad && \mathbb{E}\left[\mathcal{J}(\boldsymbol{\theta}, \mathbf{v})\right] && \text{(21a)}\\
&\text{subject to} && \mathbf{w}_{t+1} = \mathcal{G}_t\left(\mathbf{y}_{1:t}, \mathbf{w}_{1:t}\right), \;\; t = 0, \ldots, T-1, && \text{(21b)}\\
& && \mathbf{v} = \mathcal{F}\left(\mathbf{y}_{1:T}, \mathbf{w}_{1:T}\right), && \text{(21c)}
\end{aligned}$$

- Since in solving (21), we may not have access to labeled data for the desired output $\mathbf{v}$, we need to use unsupervised learning paradigm.

We propose a unified deep learning framework to handle this general formulation.

- Since the dimension of the historical observations increases as the time index t increases, using the entire history for constructing $\mathbf{w}_t$ is not scalable.
- So, it is desirable to abstract useful information from the historical observations (for the purpose of sensing vector design) into a fixed-dimensional state information vector $\mathbf{s}_t \in \mathbb{R}^S$.
- Once $\mathbf{s}_t$ is known, we can use an L-layer fully-connected DNN to map $\mathbf{s}_t$ to the optimized $\mathbf{w}_t$

$$\mathbf{w}_{t+1} = \sigma_L \left(\mathbf{A}_L \sigma_{L-1} \left(\cdots \sigma_1 \left(\mathbf{A}_1 \mathbf{s}_t + \mathbf{b}_1 \right) \cdots \right) + \mathbf{b}_L \right). \tag{22}$$

- **Question:** How to generate state variable $\mathbf{s}_t$ as function of historical observations at agent?
- We propose to use an long short-term memory (LSTM) network which is well-suited to keep track of arbitrary long-term dependencies in the input sequences.
- In particular, an LSTM cell is employed in each time frame t, which takes the new measurement $\mathbf{y}_t$ as the input vector and updates the state vectors $\mathbf{c}_t$ and $\mathbf{s}_t$ as follows:

$$\begin{aligned}
\mathbf{f}_t &= \text{sigmoid}(\mathbf{A}_\text{f}\mathbf{y}_t + \mathbf{U}_\text{f}\mathbf{s}_{t-1} + \mathbf{b}_\text{f}), \\
\mathbf{i}_t &= \text{sigmoid}(\mathbf{A}_\text{i}\mathbf{y}_t + \mathbf{U}_\text{i}\mathbf{s}_{t-1} + \mathbf{b}_\text{i}), \\
\mathbf{o}_t &= \text{sigmoid}(\mathbf{A}_\text{o}\mathbf{y}_t + \mathbf{U}_\text{o}\mathbf{s}_{t-1} + \mathbf{b}_\text{o}), \\
\mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tanh(\mathbf{A}_\text{c}\mathbf{y}_t + \mathbf{U}_\text{c}\mathbf{s}_{t-1} + \mathbf{b}_\text{c}), \\
\mathbf{s}_t &= \mathbf{o}_t \circ \tanh(\mathbf{c}_t).
\end{aligned}$$

$\mathbf{f}_t$: Forget gate's activation vector,
$\mathbf{i}_t$: Input/update gate's activation vector,
$\mathbf{o}_t$: Output gate's activation vector,
$\mathbf{c}_t$: Cell state vector,
$\mathbf{s}_t$: Hidden state vector.

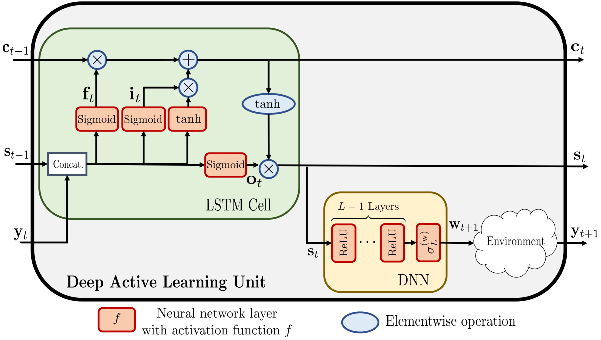
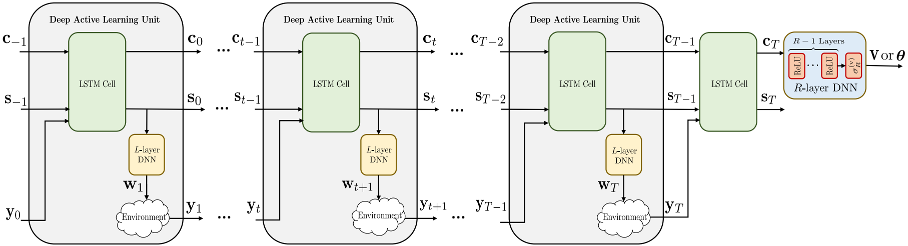


Fig.: The proposed deep active learning unit for designing the next sensing vector $\mathbf{w}_{t+1}$ and updating the cell state vector $\mathbf{c}_t$ as well as the hidden state vector $\mathbf{s}_t$, given the new measurement $\mathbf{y}_t$ and the previous state vectors $\mathbf{c}_{t-1}$ and $\mathbf{s}_{t-1}$.

- **Question:** How to train this DNN?

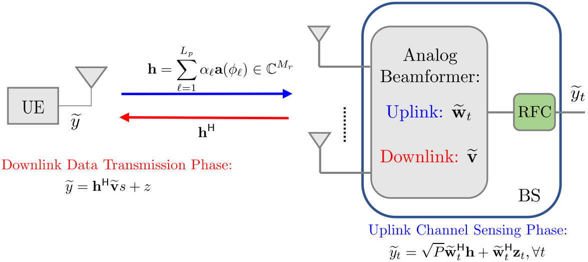
Deep Active Learning Framework



- By considering all sensing stages, we can think of the proposed end-to-end architecture as a very deep neural network.
- The ultimate goal of this network is either to estimate the system parameter $\boldsymbol{\theta}$ or to design $\mathbf{v}$ by maximizing the utility function $\mathcal{J}(\boldsymbol{\theta}, \mathbf{v})$
- Accordingly, we employ another DNN in time frame T to map the final cell state vector $\mathbf{c}_T$ to the estimate of $\boldsymbol{\theta}$ or to the design of $\mathbf{v}$ for the general problem (21) as:

$$\hat{\boldsymbol{\theta}} \text{ or } \mathbf{v} = \widetilde{\sigma}_{\tilde{L}} \left(\widetilde{\mathbf{A}}_{\tilde{L}} \widetilde{\sigma}_{\tilde{L}-1} \left(\cdots \widetilde{\sigma}_1 \left(\widetilde{\mathbf{A}}_1 \mathbf{c}_T + \widetilde{\mathbf{b}}_1 \right) \cdots \right) + \widetilde{\mathbf{b}}_{\tilde{L}} \right), \tag{23}$$

- We train the overall DNN by employing an SGD algorithm in order to minimize the empirical average MSE or to maximize average utility function for problem (21).



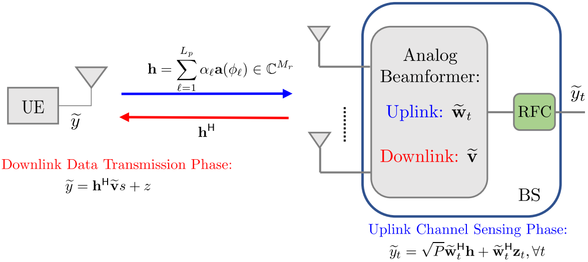
How to adaptively design uplink sensing $\{\widetilde{\mathbf{w}}_t\}_{t=1}^{T}$ to estimate AoAs $\{\phi_\ell\}_{\ell=1}^{L_p}$, or to maximize the downlink beamforming gain $|\mathbf{h}^{\mathsf{H}}\widetilde{\mathbf{v}}|^2$?

- BS with M_r antennas and a single RF chain serves a single-antenna user in mmWave band.
- **Problem Formulation I:** Minimizing the AoA estimation error by designing an adaptive uplink sensing strategy, i.e.,

$$\underset{\{\widetilde{\mathcal{G}}_t(\cdot,\cdot)\}_{t=0}^{T-1},\, \widetilde{\mathcal{F}}(\cdot,\cdot)}{\text{minimize}} \quad \mathbb{E}\left[\|\hat{\phi} - \phi\|_2^2\right] \tag{24a}$$

$$\text{subject to} \quad \widetilde{\mathbf{w}}_{t+1} = \widetilde{\mathcal{G}}_t\left(\widetilde{y}_{1:t}, \widetilde{\mathbf{w}}_{1:t}\right), \; t = 0, \ldots, T-1, \tag{24b}$$

$$\hat{\phi} = \widetilde{\mathcal{F}}\left(\widetilde{y}_{1:T}, \widetilde{\mathbf{w}}_{1:T}\right). \tag{24c}$$



How to adaptively design uplink sensing $\{\widetilde{\mathbf{w}}_t\}_{t=1}^{T}$ to estimate AoAs $\{\phi_\ell\}_{\ell=1}^{L_p}$, or to maximize the downlink beamforming gain $|\mathbf{h}^{\mathsf{H}}\widetilde{\mathbf{v}}|^2$?

- **Problem Formulation II:** Maximizing the downlink beamforming gain (under some precoding constraint) based on the uplink CSI learned in the adaptive sensing phase, i.e.,

$$\underset{\{\widetilde{\mathcal{G}}_t(\cdot,\cdot)\}_{t=0}^{T-1}, \widetilde{\mathcal{F}}(\cdot,\cdot)}{\text{maximize}} \quad \mathbb{E}\left[|\mathbf{h}^{\mathsf{H}}\widetilde{\mathbf{v}}|^2\right] \tag{25a}$$

$$\text{subject to} \quad \widetilde{\mathbf{w}}_{t+1} = \widetilde{\mathcal{G}}_t\left(\widetilde{y}_{1:t}, \widetilde{\mathbf{w}}_{1:t}\right), \ t = 0, \ldots, T-1, \tag{25b}$$

$$\mathbf{v} = \widetilde{\mathcal{F}}\left(\widetilde{y}_{1:T}, \widetilde{\mathbf{w}}_{1:T}\right), \tag{25c}$$

- Both of these problem formulations are in the form of generic active sensing problems discussed earlier. So, the proposed deep active learning approach can be used.

- **Implementation platform:** TensorFlow and Keras.
- **Optimization method:** Adam optimizer with an adaptive learning rate initialized to 0.001.
- **Deep Active Learning Unit:**
  - LSTM cell where the dimension of the hidden state is $S = 512$.
  - 4-layer DNN with dense layers of widths $[1024, 1024, 1024, 2M_r]$ for sensing vector.
  - Rectified linear unit (ReLU) as the activation function of the hidden layers.
  - Proper normalization layer as the activation function of the last layer to ensure the power constraint is satisfied.
- **Final DNN:**
  - Rectified linear unit (ReLU) as the activation function of the hidden layers.
  - AoA Estimation Problem: 2-layer network with widths $[512, L_p]$. The last layer is linear.
  - BF Gain Maximization Problem: 4-layer network with widths $[1024, 1024, 1024, 2M_r]$, while the last layer is a proper normalization layer to satisfy the power constraint.

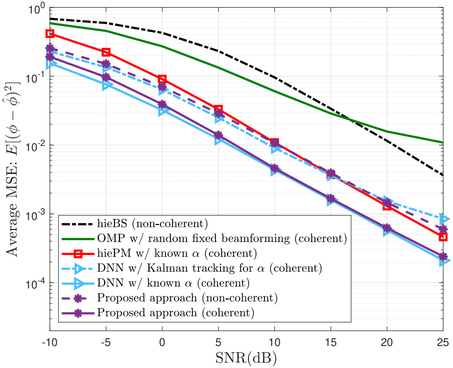
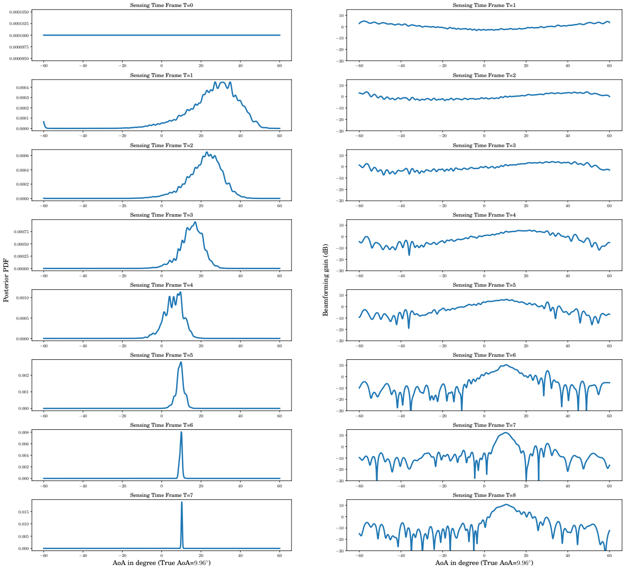


Fig.: Average MSE versus SNR for different beam alignment methods in a system with $M_r = 64$, $L_p = 1$, $T = 14$, and $\phi \in [-60^\circ, 60^\circ]$. In this experiment, the sensing vectors must satisfy the 2-norm constraint.

- **Baselines:**
  - Compressive sensing with fixed beamforming, e.g., OMP [Tropp, Gilbert, 2007].
  - Hierarchical codebook with bisection search (hieBS) [Alkhateeb, Ayach, Leus, Heath, 2014].
  - Hierarchical codebook with posterior matching (hiePM) [Chiu, Ronquillo, Javidi, 2019],
  - Codebook-free posterior based DNN (DNN) [Sohrabi, Chen, Yu, 2021].
- The proposed method can also tackle the noncoherent sensing setup in which the phase information of the measurement signal is not available.

Posterior Distribution of AoA and Optimized Sensing Vectors



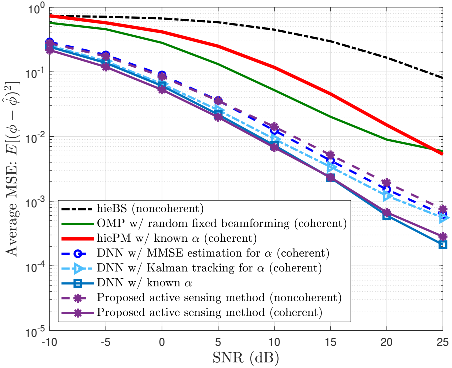


Fig.: Average MSE versus SNR for different beam alignment methods in a system with $M_r = 64$, $L_p = 1$, $T = 14$, and $\phi \in [-60^\circ, 60^\circ]$. In this experiment, the sensing vectors satisfy the constant modulus constraint.

- **Baselines:**
  - Compressive sensing with fixed beamforming, e.g., OMP [Tropp, Gilbert, 2007].
  - Hierarchical codebook with bisection search (hieBS) [Alkhateeb, Ayach, Leus, Heath, 2014].
  - Hierarchical codebook with posterior matching (hiePM) [Chiu, Ronquillo, Javidi, 2019].
  - Codebook-free posterior based DNN (DNN) [Sohrabi, Chen, Yu, 2021].
- The proposed method can also tackle the noncoherent sensing setup in which the phase information of the measurement signal is not available.

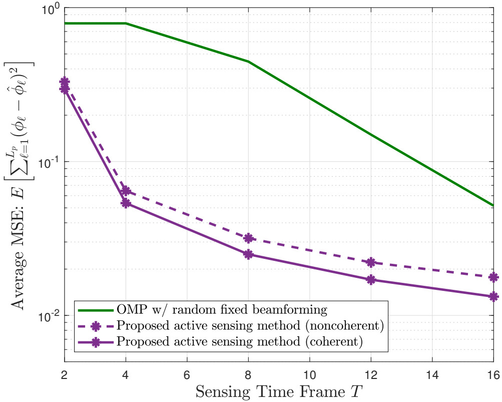


Fig.: Average MSE versus sensing time frames T for two AoAs in a system with $M_r = 64$, $L_p = 2$, SNR $= 25$dB, and $\phi_1, \phi_2 \in [-60^\circ, 60^\circ]$. In this experiment, the sensing vectors must satisfy the 2-norm constraint.

Downlink Beamformer Design

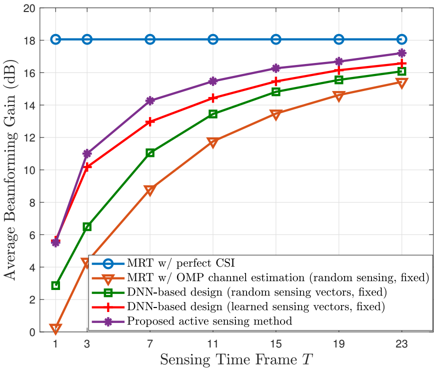
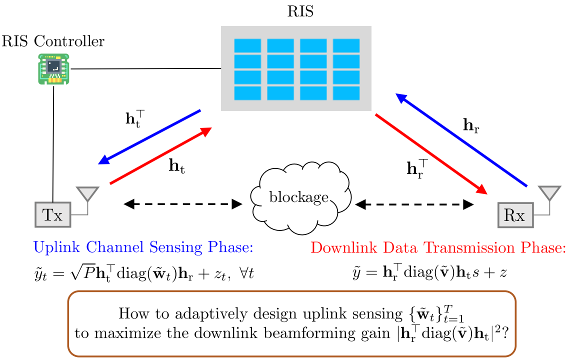


Fig.: Average beamforming gain in dB versus sensing time frames T for different methods in a system with $M_r = 64$, SNR $= 0$dB, $L_p = 3$, and $\phi_1, \phi_2, \phi_3 \in [-60^\circ, 60^\circ]$. In this experiment, the sensing vectors of each method satisfy the 2-norm constraint.

- **Baselines:**
  - Maximum-ratio transmission (MRT) with perfect CSI (performance upper bound).
  - MRT with compressed sensing CSI estimation, e.g., OMP [Tropp, Gilbert, 2007].
  - DNN-based design with nonadaptive sensing [Attiah, Sohrabi, Yu, 2020].

Application to

Intelligent Reflecting Surface



- **Problem Formulation:** Maximizing the downlink beamforming gain of an RIS-assisted system based on the uplink CSI learned in the active sensing phase, i.e.,

$$\underset{\{\tilde{\mathcal{G}}_t(\cdot,\cdot)\}_{t=0}^{T-1},\tilde{\mathcal{F}}(\cdot,\cdot)}{\text{maximize}} \quad \mathbb{E}\left[|\mathbf{h}_{\mathrm{c}}^{\top}\tilde{\mathbf{v}}|^2\right] \tag{26a}$$

$$\text{subject to} \quad \tilde{\mathbf{w}}_{t+1} = \tilde{\mathcal{G}}_t(\tilde{y}_{1:t}, \tilde{\mathbf{w}}_{1:t}), \ t = 0, \ldots, T-1, \tag{26b}$$

$$\tilde{\mathbf{v}} = \tilde{\mathcal{F}}(\tilde{y}_{1:T}, \tilde{\mathbf{w}}_{1:T}), \tag{26c}$$

where $\mathbf{h}_{\mathrm{c}} \triangleq \mathrm{diag}(\mathbf{h}_{\mathrm{t}})\mathbf{h}_{\mathrm{r}}$ is the cascaded effective channel between Tx and Rx.

Implementation Details

- **Implementation platform:** TensorFlow and Keras.
- **Optimization method:** Adam optimizer with an adaptive learning rate initialized to 0.001.
- **Deep Active Learning Unit:**
  - LSTM cell where the dimension of the hidden state is $S = 512$.
  - 4-layer DNN with dense layers of widths $[1024, 1024, 1024, 2M_r]$ for sensing vector.
  - Rectified linear unit (ReLU) as the activation function of the hidden layers.
  - Proper normalization layer as the activation function of the last layer to ensure the RIS constant modulus constraint is satisfied.
- **Final DNN:**
  - Rectified linear unit (ReLU) as the activation function of the hidden layers.
  - 4-layer network with widths $[1024, 1024, 1024, 2M_r]$, while the last layer is a proper normalization layer to satisfy the RIS constraint.
- **Channel Model:** Rician fading channel with Rician factor $\varepsilon = 10$, for which $\mathbf{h}_{\mathrm{t}}$ is given by:

$$\mathbf{h}_{\mathrm{t}} = \sqrt{\frac{\varepsilon}{1+\varepsilon}}\tilde{\mathbf{h}}_{\mathrm{t}}^{\mathrm{LOS}} + \sqrt{\frac{1}{1+\varepsilon}}\tilde{\mathbf{h}}_{\mathrm{t}}^{\mathrm{NLOS}}, \tag{27}$$

  where $\tilde{\mathbf{h}}_{\mathrm{t}}^{\mathrm{LOS}} = \tilde{\alpha}\,\tilde{\mathbf{a}}(\theta_{\mathrm{t}}, \phi_{\mathrm{t}})$ and $\tilde{\mathbf{h}}_{\mathrm{t}}^{\mathrm{NLOS}} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I})$.
- **RIS Configuration:** 8×8 rectangular array.

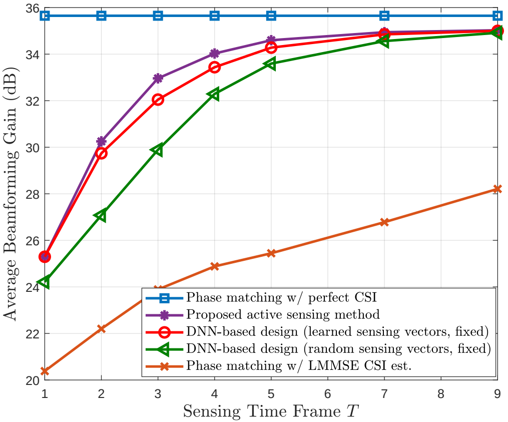


Fig.: Average beamforming gain in dB versus sensing time frames T for different methods in a system with $N_r = 64$ RIS elements, $\varepsilon = 10$, and SNR $= 0$dB.

- **Baselines:**
  - Phase matching with perfect CSI (performance upper bound).
  - Phase matching with LMMSE CSI estimation.
  - DNN-based design with fixed sensing vector.
    - The RIS coefficients in the sensing phase are set randomly [Jiang, Cheng, Yu, 2021].
    - The RIS coefficients in the sensing phase are learned using the DL framework.

Conclusions

- Sensing will be an integral part of future wireless communication systems.
- Active sensing is an important problem both theoretically and in practice.
- Deep learning can now tackle complex mathematical optimization problems more efficiently than traditional optimization approaches.
- Deep learning gives a viable data-driven methodology for designing active sensing strategies:
  - Active learning for initial alignment in mmWave communications;
  - Active sensing matrix design for reflection alignment in RIS systems.
- Key to successful application of machine learning to sensing and communications:
  - Learning objective should match the application objective.
  - It is beneficial to bypass explicit channel modeling and channel estimation.
  - Neural network architecture should match the problem structure (e.g., LSTM).

Further Information

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, vol.39, no.8, pp.2347-2360, August 2021.
https://arxiv.org/abs/2012.13607

Foad Sohrabi, Tao Jiang, Wei Cui, and Wei Yu,
"Active Sensing for Communications via Learning",
To appear in *IEEE Journal on Selected Areas in Communications, Special Issue on Integrated Sensing and Communication*, 2022.
https://arxiv.org/abs/2112.04075