Learning the Physical Layer in FDD Systems: Centralized Learning of Distributed Functions

Wolfgang Utschick

Technical University of Munich
School of Computation, Information and Technology
Munich Data Science Institute
A note on UL–DL Distributional Invariance
Distributional Invariance of the Uplink & Downlink

Invariance of
- × instantaneous UL & DL realizations (FDD systems ⇒ no reciprocity in general)
- ✓ UL & DL distributions

Distributional Invariance of the Uplink & Downlink:

“Sampling channel state information from the same propagation environment in different frequency bands, but still with similar radio propagation characteristics, represents approximately the same underlying probability distribution.”

Proof:
Not a rigorous proof, but there is some evidence by statistical hypothesis testing based on two-sample tests, cf. Utschick et al, IEEE T-WC, 2022.
An intuitive explanation (1)

Consider the simple case of a ULA with $N$ antennas. In this case the channel vector $h$ is such that:

$$h \propto [\alpha^0, \alpha^1, \alpha^2, \ldots, \alpha^{N-1}]$$

with $\alpha = \exp \left( -j \frac{2\pi df}{c} \sin \theta \right)$
An intuitive explanation (2)

At the same time, for a different carrier frequency we have...

\[ \frac{c}{(f + \delta f)} \]

\[ \theta + \delta \theta \]

\[ d \]
Small changes $\delta f$ in the carrier frequency can be compensated by small changes $\delta \theta$ of $\theta$, therefore $h$ is not changed if

$$f \sin \theta = (f + \delta f) \sin (\theta + \delta \theta)$$
**Definition:** Given a positive definite kernel \( k(\cdot, \cdot) = \langle \varphi(\cdot), \varphi(\cdot) \rangle \) of a reproducing kernel Hilbert space (RKHS) \( \mathcal{H}_k \) with a feature map \( \varphi(\cdot) \in \mathcal{H}_k \), the maximum mean discrepancy (MMD) between two probability distributions \( \mathbb{P} \) and \( \mathbb{Q} \) can be obtained by

\[
\text{MMD}^2(\mathbb{P}, \mathbb{Q}, k) = \mathbb{E}[k(p, p') + k(q, q') - 2k(p, q)],
\]

with random variables \( (p, q) \sim \mathbb{P} \times \mathbb{P} \) and \( (q', p') \sim \mathbb{Q} \times \mathbb{Q} \). It follows that \( \text{MMD}(\mathbb{P}, \mathbb{Q}, k) = 0 \) if and only if \( \mathbb{P} = \mathbb{Q} \). We further assume that we have sample sets \( \mathcal{P} \sim \mathbb{P} \) and \( \mathcal{Q} \sim \mathbb{Q} \) of equal sample size \( n \), an unbiased estimator of the squared MMD for measuring the discrepancy between \( \mathbb{P} \) and \( \mathbb{Q} \) can be obtained as

\[
\text{MMD}^2(\mathbb{P}, \mathbb{Q}, k) = \frac{1}{n(n-1)} \sum_{i \neq j} h_{ij},
\]

where \( h_{ij} = k(p_i, p_j) + k(q_i, q_j) - k(p_i, q_j) - k(p_j, q_i) \) with \( p_i \in \mathcal{P} \) and \( q_i \in \mathcal{Q} \) being the realizations of the random variables \( p \sim \mathbb{P} \) and \( q \sim \mathbb{Q} \). Following the usual kernel trick, we swap the choice of feature map \( \varphi(\cdot) \) with the decision for the kernel function \( k(\cdot, \cdot) \). The most common choice for a kernel is the Gaussian kernel, i.e.,

\[
k(p, q) = \exp \left( - \frac{\|p - q\|^2}{\sigma^2} \right),
\]

where \( p \in \mathcal{P} \) and \( q \in \mathcal{Q} \) are two samples drawn from \( \mathbb{P} \) and \( \mathbb{Q} \) and \( \sigma^2 \) corresponds to the 50-percentile (median) distance between elements in the aggregate sample, as suggested in [28].
Hypothesis Testing Based On Two-Sample Tests


Why is this important?
Because what really matters in machine learning is the distribution of data!
Because what really matters in machine learning is the distribution of data!
Novel Design Options for FDD PHY Layer Functions
PHY Layer Functions Soley Based on UL-Data

- **BS**: collect UL data
- **DNN**: learn $f_{DNN}(\cdot)$, share parameters
- **MT**: feedback, CSI, etc.
- **Observation $Y$**

Mathematical notation:
- $f_{DNN}(\cdot)$
- $\mathcal{H}^{UL}$

Diagram:
- BS to DNN: collect UL data
- BS to DNN: share parameters
- DNN to DNN: observation $Y$
- DNN to MT: feedback, CSI, etc.
Application:

Channel Compression & Reconstruction
1. Train an autoencoder solely based on UL CSI

\[ \tilde{H}_{UL}@BS \xrightarrow{\text{Enc}} Enc \rightarrow Dec \rightarrow H_{UL}@BS \]

- \( f_\theta(\cdot) \): encoder neural net
- \( g_\phi(\cdot) \): decoder neural net

\[ z_{UL} = f_\theta(\tilde{H}_{UL}) \]

\[ H_{UL} \approx \hat{H}_{UL} = g_\phi(z_{UL}). \]
Channel Compression & Reconstruction

2. Offload the UL-trained encoder to each MTs in the cell
3. Compress the DL CSI at the MT

\[ z_{DL} = f_\theta(\tilde{H}_{DL}) \]
4. Feed the encoded DL CSI back to the BS
5. Decode the DL CSI at the BS with the UL-trained decoder

\[ H_{DL} \approx \hat{H}_{DL} = g_\phi(z_{DL}) \]
Channel Compression & Reconstruction

\[ \tilde{H}_{UL}@BS \xrightarrow{\text{Enc}} f_\theta(\cdot) \xrightarrow{\text{Dec}} H_{UL}@BS \]

\[ \tilde{H}_{DL}@MT \xrightarrow{\text{Enc}} g_\phi(\cdot) \xrightarrow{\text{Dec}} H_{DL}@BS \]

- not necessary to collect DL CSI data for training
- robust against Gaussian noise

### Channel Compression & Reconstruction

<table>
<thead>
<tr>
<th>Layer type</th>
<th>Output shape</th>
<th>#Parameters $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>64 x 160 x 2</td>
<td>0</td>
</tr>
<tr>
<td>Conv2D, strides=2</td>
<td>32 x 80 x 8</td>
<td>152</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>32 x 80 x 8</td>
<td>32</td>
</tr>
<tr>
<td>ReLU</td>
<td>32 x 80 x 8</td>
<td>0</td>
</tr>
<tr>
<td>Conv2D, strides=2</td>
<td>16 x 40 x 16</td>
<td>1168</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>16 x 40 x 16</td>
<td>64</td>
</tr>
<tr>
<td>ReLU</td>
<td>16 x 40 x 16</td>
<td>0</td>
</tr>
<tr>
<td>Conv2D, strides=2</td>
<td>8 x 20 x 32</td>
<td>4640</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>8 x 20 x 32</td>
<td>128</td>
</tr>
<tr>
<td>ReLU</td>
<td>8 x 20 x 32</td>
<td>0</td>
</tr>
<tr>
<td>Conv2D, strides=2</td>
<td>4 x 10 x 64</td>
<td>18496</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>4 x 10 x 64</td>
<td>256</td>
</tr>
<tr>
<td>ReLU</td>
<td>4 x 10 x 64</td>
<td>0</td>
</tr>
<tr>
<td>Conv2D, strides=2</td>
<td>2 x 5 x 128</td>
<td>73856</td>
</tr>
<tr>
<td>Batch normalization</td>
<td>2 x 5 x 128</td>
<td>512</td>
</tr>
<tr>
<td>ReLU</td>
<td>2 x 5 x 128</td>
<td>0</td>
</tr>
<tr>
<td>Flatten</td>
<td>1280</td>
<td>0</td>
</tr>
<tr>
<td>Fully-connected</td>
<td>256</td>
<td>327936</td>
</tr>
<tr>
<td>Tanh</td>
<td>256</td>
<td>0</td>
</tr>
</tbody>
</table>

- autoencoder is based on convolutional layers instead of fully connected layers
  $\implies$ architecture can be easily scaled to high dimensional CSI
Channel Compression & Reconstruction

- QuaDRiGa channel simulator
- Urban Microcell NLoS
- 58 paths
- training samples = 48 K
- validation/test samples = 2 × 6 K
- number of antennas = 64
- number of carriers = 160
- center frequency = 2.5 GHz
- frequency gap = 120 and 480 MHz

Figure 1. UMi NLoS

4. Results intro
This report illustrates the results that can be submitted for the Journal publication. In particular, the following points are addressed here:

- The full cell is considered. This means that we removed the restriction on having extremely high density.
- We compare different frequency gaps. Namely:
  - We see how the MMD changes with different gaps,
  - how the null-hypothesis testing performs with different gaps
- CDF and pdfs of the NMSE and Cosine similarity are shown, together with the average values
- We will also test the trained CNN on another cell to see how well it generalizes.

Before that it is important that a few comments are made on QuaDRiGa.

5. Comment on QuaDRiGa channels
QuaDRiGa does not have a proper solution for FDD systems and offers two options for generating channels: single-frequency and multi-frequency.

So far, the former has been used because of the small gap (120 MHz) we had. The idea of this approach is to consider a larger bandwidth and then to cut the portion of the bandwidth that corresponds to the frequency gap.

However, what QuaDRiGa recommends for larger gaps is to use the so-called multi-frequency approach. Here we have to specify as many center frequencies as we wish and for a given environment we have:

- same positions for Base Station and Mobile Station
- cluster delays and angles for each multi-path component are the same
- Spatial consistency of the large scale fading parameters is identical.

On the other hand for each frequency we have:

- Path-loss is different for each frequency
- Path-powers are different for each frequency
- Delay and angular spreads are different
- K-Factor is different
- Cross-polarization ratio (XPR) of the NLOS components is different.

These cause the channels to have different impulse responses at each frequency, due to the fact that despite the environment is the same the channels change because of the randomness induced by the small scale fading parameters and because of the parameters that change with the frequency.

However, for our task we can still use the multi-frequency approach because we are not training UL-DL end to end and therefore we do not need the exact pairs. Because of the randomness induced by the small scale fading parameters, with the multi-frequency approach the channels differ at the same frequency as well.

Figure ?? and ?? one can see the difference in the antenna pattern between a ULA antenna and a 3GPP-3d antenna.
Channel Compression & Reconstruction

- QuaDRiGa channel simulator
- Urban Microcell NLoS
- 58 paths
- training samples = 48 K
- validation/test samples = 2 × 6 K
- number of antennas = 64
- number of carriers = 160
- center frequency = 2.5 GHz
- frequency gap = 120 and 480 MHz

Figure 1. UMi NLoS

4. Results intro

This report illustrates the results that can be submitted for the Journal publication. In particular, the following points are addressed here:

- The full cell is considered. This means that we removed the restriction on having extremely high density.
- We compare different frequency-gap. Namely:
  - We see how the MMD changes with different gaps,
  - how the null-hypothesis testing performs with different gaps
- CDF and pdfs of the NMSE and Cosine similarity are shown, together with the average values
- We will also test the trained CNN on another cell to see how well it generalizes.

Before that it is important that a few comments are made on QuaDRiGa.

5. Comment on QuaDRiGa channels

QuaDRiGa does not have a proper solution for FDD systems and offers two options for generating channels: single-frequency and multi-frequency.

So far, the former has been used because of the small gap (120 MHz) we had. The idea of this approach is to consider a larger bandwidth and then to cut the portion of the bandwidth that corresponds to the frequency gap.

However, what QuaDRiGa recommends for larger gaps is to use the so-called multi-frequency approach. Here we have to specify as many center frequencies as we wish and for a given environment we have:

- same positions for Base Station and Mobile Station
- cluster delays and angles for each multi-path component are the same
- Spatial consistency of the large scale fading parameters is identical.

On the other hand for each frequency we have:

- Path-loss is different for each frequency
- Path-powers are different for each frequency
- Delay and angular spreads are different
- K-Factor is different
- Cross-polarization ratio (XPR) of the NLOS components is different.

These cause the channels to have different impulse responses at each frequency, due to the fact that despite the environment is the same the channels change because of the randomness induced by the small scale fading parameters and because of the parameters that change with the frequency.

However, for our task we can still use the multi-frequency approach because we are not training UL-DL end to end and therefore we do not need the exact pairs. Because of the randomness induced by the small scale fading parameters, with the multi-frequency approach the channels differ at the same frequency as well.

Figure 2. UMi NLoS
Channel Compression & Reconstruction

NMSE of different methods for SNR=10 dB.

CDF($\varepsilon^2$ dB)
Channel Compression & Reconstruction

- achievable per-user rate in a multiuser scenario (8 user)
- zero-forcing precoding based on recovered DL CSI
- the hyperbolic tangent as activation function in the latent space makes easy to quantize the CSI
Application:

Codebook Construction & Feedback Generation
Proposed concept: Training based soley on uplink (UL) data at the base station (@BS) and subsequent offloading of trained deep neural networks to the mobile terminals (MTs) for feedback generation, channel estimation, channel compression, etc.

- kind of distributed implementation of AI-aided physical layer functions
- offloading deep functionalities anywhere and anytime in the networks for increasing performance

Urban Macrocell Scenario

3GPP 38.901 UMa, single carrier scenario:
Urban Macrocell Scenario

3GPP 38.901 UMa, single carrier scenario:

Distributional invariance holds here!
Urban Macrocell Scenario

3GPP 38.901 UMa, single carrier scenario:

- QuaDRiGa channel simulator
- MIMO channels: (BS antennas, MT antennas) = (16,4) or (32,16), non-line-of-sight (NLOS), line-of-sight (LOS) and outdoor-to-indoor (O2I)
- UL carrier frequency = 2.53 GHz,
- DL carrier frequency = 2.73 GHz,
- BS-ULA with “3GPP-3D” antennas,
- MT-ULA with ”omni- directional” antennas,
- BS placed at a height of 25 m with a sector of 120°,
- minimum distance of the MT location to the BS is 35 m,
- maximum distance to the BS is 500 m.
- $10^4$ training, $2.5 \times 10^3$ validation and $\sim 5 \times 10^3$ test samples.
Codebook Construction
Codebook Construction

- unsupervised codebook design
- using the k-means algorithm based on the achievable date rate metric
- solely based on UL channel state information (CSI)

1. Divide the training set $\mathcal{H}$ into $K$ clusters $\mathcal{V}_k^{(i)}$:

$$\mathcal{V}_k^{(i)} = \{ H \in \mathcal{H} | r(H, Q_k^{(i)}) \geq r(H, Q_j^{(i)}), k \neq j \}.$$  

2. Find new covariance matrices or update the so called “cluster centers”:

$$Q_{k}^{(i+1)} = \underset{Q \succeq 0}{\arg \max} \frac{1}{|\mathcal{V}_k^{(i)}|} \sum_{H \in \mathcal{V}_k^{(i)}} r(H, Q)$$

subject to \( trace(Q) \leq \rho \) and \( rank(Q) \leq N_{rx} \).

Applying a projected gradient algorithm on the candidate set of precoding covariances:

$$g_Q = \frac{1}{\sigma_n^2 \ln(2)} \sum_{H \in \mathcal{V}_k^{(i)}} H^H \left( I + \frac{1}{\sigma_n^2} HQH^H \right)^{-1} H,$$

$$Q \leftarrow Q + \alpha g_Q.$$
$N_{tx} = 16$, $N_{rx} = 4$, with different codebook sizes ($2^M$)
Application:
Channel Estimation
Channel Estimation

Downlink observations @MT:

\[ Y_{i,DL} = H_{i,DL}X + Z_i \]

Emulated downlink system @BS:

\[ \tilde{Y}_{i,DL} = H_{i,UL}^TX + Z_i \]

Channel Estimation

$N_{\text{BS}} = 64$, $N_{\text{MS}} = 4$, $N_{\text{P}} = 64$, mixed NLOS/LOS scenario, 20K training samples
Channel Estimation

$N_{BS} = 64, N_{MS} = 4, N_P = 64$, mainly LOS scenario, 20K training samples

![Graph showing Channel Estimation results with SNR in dB on the x-axis and Normalized MSE on the y-axis. The graph compares different estimation methods including LMMSE (DL), LMMSE (UL), LS, ML, genie OMP, AI (DL), AI (UL), AI (UMa).](image-url)
Thank You!
Links to Recent Work

- Channel Estimation, Prediction, and Extrapolation:

- Codebook Design and Feedback Generation in Multiuser MIMO Systems:
Appendix
Quantitative Analysis:
More About Two-Sample Tests
MMD definition

The two-sample tests are based on the maximum mean discrepancy metric. The **maximum mean discrepancy** is the distance between feature means:

$$\text{MMD}^2(P, Q, k) := E[k(p, p') + k(q, q') - 2k(p, q)],$$  \hspace{1cm} (1)

where:

- $k(\cdot, \cdot) = \langle \varphi(\cdot), \varphi(\cdot) \rangle$: positive definite kernel of a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$ with a feature map $\varphi(\cdot) \in \mathcal{H}_k$
- $(p, p') \sim P \times P$
- $(q, q') \sim Q \times Q$

Hence, $\text{MMD}(P, Q, k) = 0$ if and only if $P = Q$.  


Kernels descriptions

- **“1-percentile”**: $\sigma = 1$-percentile of the distance between points in the aggregate sample
  \[ k(a, b) = \exp \left( -\frac{\|a - b\|^2}{\sigma^2_1} \right) \]

- **“median”**: $\sigma = 50$-percentile of the distance between points in the aggregate sample
  \[ k(a, b) = \exp \left( -\frac{\|a - b\|^2}{\sigma^2_{50}} \right) \]

- **“deep”**: $\sigma = 50$-percentile of the distance between points in the aggregate sample of the latent space of an autoencoder
  \[ k(a, b) = \exp \left( -\frac{\|f_{\theta}(a) - f_{\theta}(b)\|^2}{\sigma^2_{50,\text{latent}}} \right) \]
Different frequency gaps in a UMi LOS scenario

The larger the frequency gaps, the larger is the discrepancy between the UL & DL distributions.
Over different UMi NLOS ($\Delta f = 120$ MHz)

**Median**

- **same seed**
  - Median $\approx -0.2$
  - Median $\approx -0.4$

- **seed 10**
  - Median $\approx -0.2$
  - Median $\approx -0.4$

- **seed 27**
  - Median $\approx -0.2$
  - Median $\approx -0.4$

- **seed 130**
  - Median $\approx -0.2$
  - Median $\approx -0.4$

**1-percentile**

- **same seed**
  - 1-percentile $\approx -0.1$
  - 1-percentile $\approx -0.2$

- **seed 10**
  - 1-percentile $\approx -0.1$
  - 1-percentile $\approx -0.2$

- **seed 27**
  - 1-percentile $\approx -0.1$
  - 1-percentile $\approx -0.2$

- **seed 130**
  - 1-percentile $\approx -0.1$
  - 1-percentile $\approx -0.2$

**Deep**

- **same seed**
  - Deep $\approx -0.1$
  - Deep $\approx -0.2$

- **seed 10**
  - Deep $\approx -0.1$
  - Deep $\approx -0.2$

- **seed 27**
  - Deep $\approx -0.1$
  - Deep $\approx -0.2$

- **seed 130**
  - Deep $\approx -0.1$
  - Deep $\approx -0.2$