Speeding up Training of Linear Predictors for Multi-Antenna Frequency-Selective Channels via Meta-Learning

Sangwoo Park * and Osvaldo Simeone

Department of Engineering, King's College London, London WC2R 2LS, UK
* Correspondence: sangwoo.park@kcl.ac.uk

Abstract: An efficient data-driven prediction strategy for multi-antenna frequency-selective channels must operate based on a small number of pilot symbols. This paper proposes novel channel-prediction algorithms that address this goal by integrating transfer and meta-learning with a reduced-rank parametrization of the channel. The proposed methods optimize linear predictors by utilizing data from previous frames, which are generally characterized by distinct propagation characteristics, in order to enable fast training on the time slots of the current frame. The proposed predictors rely on a novel long short-term decomposition (LSTD) of the linear prediction model that leverages the disaggregation of the channel into long-term space-time signatures and fading amplitudes. We first develop predictors for single-antenna frequency-flat channels based on transfer/meta-learned quadratic regularization. Then, we introduce transfer and meta-learning algorithms for LSTD-based prediction models that build on equilibrium propagation (EP) and alternating least squares (ALS). Numerical results under the 3GPP 5G standard channel model demonstrate the impact of transfer and meta-learning on reducing the number of pilots for channel prediction, as well as the merits of the proposed LSTD parametrization.

Keywords: channel prediction; meta-learning; multi-antenna frequency-selectivity; equilibrium propagation

1. Introduction

The capacity to accurately predict channel state information (CSI) is a key enabler of proactive resource allocation strategies, which are central to many visions for efficient and low-latency communications in 6G and beyond (see, e.g., [1]). The problem of channel prediction is relatively straightforward in the presence of known channel statistics. In fact, under the common assumption that multi-antenna frequency-selective channels follow stationary complex Gaussian processes, optimal channel predictors can be obtained via linear minimum mean squared error (LMMSE) estimators such as the Wiener filter [2]. However, in practice, the channel statistics are not known, and predictors need to be optimized based on training data obtained through the transmission of pilot signals [3–8]. The problem addressed by this paper concerns the design of data-efficient channel predictors for multi-antenna frequency-selective channels.

1.1. Context and Prior Art

A classical approach to tackle this problem is to optimize finite impulse response (FIR) filters [9], or recursive linear filters via autoregressive (AR) models [3–5] such as Kalman filtering (KF) [6–8], by estimating channel statistics from the available pilot data. Although recursive linear filters generally outperform FIR filters when an accurate model of the state-transition dynamics is available [10,11], FIR filters are typically advantageous in the presence of limited amounts of pilot data [12]. More recently, deep learning-based nonlinear predictors have also been proposed to adapt to channel statistics through the
training of neural networks, namely recurrent neural networks [13–16], convolutional neural networks [17,18], and multi-layer perceptrons [19].

As reported in [14–16,19], deep learning-based predictors tend to require larger training (pilot) data, and fail to outperform well-designed linear filters in the low-data regime. Some solutions addressing this issues include [20], which applies reinforcement learning to determine whether to predict channels or not at the current time, and the use of hypernetworks to adapt parameters of a KF accordingly to current channel dynamics [12].

Most prior work, with the notable exception of [12], focuses on the optimization of channel predictors under the assumption of a stationary spatio-temporal correlation function across the time interval of interest. This conventional approach fails to leverage common structure that may exist across multiple frames, with each frame being characterized by distinct spatio-temporal correlations (see Figure 1). Reference [12] allowed for varying Doppler spectra across frames, through a deep learning-based hypernetwork that is used to adapt the parameters of a generative model [12].

![Illustration of the frame-based transmission system under study. At any frame f, based on the previous N channels h_{i,f}, we investigate the problem of optimizing the δ-lag prediction \( \hat{h}_{i+\delta,f} \).](image)

This paper takes a different approach that allows us to move beyond the single-antenna setting studied in [12]. As described in the next subsection, key ingredients of the proposed methods are transfer learning and meta-learning. Transfer learning [21] and meta-learning [22] aim at using knowledge from distinct tasks in order to reduce the data requirements on a new task of interest. Given a large enough resemblance between different tasks, both transfer learning and meta-learning have shown remarkable performance to reduce the sample complexity in general machine learning problems [23]. Transfer learning applies to a specific target task, whereas meta-learning caters to adaptation to any new task (see e.g., [24]).

Previous applications of transfer learning to communication systems include beamforming for multi-user, multiple-input, single-output (MISO) downlink [25] and for intelligent reflecting surfaces (IRS)-assisted MISO downlink [26], and downlink channel prediction [27,28] (see also [25,27]). Meta-learning has been applied to communication systems, including demodulation [29–32], decoding [33], end-to-end design of encoding and decoding with and without a channel model [34,35]; MIMO detection [36], beamforming for multiuser MISO downlink systems via [37], layered division multiplexing for ultra-reliable communications [38], UAV trajectory design [39], and resource allocation [40].

1.2. Contributions

This paper proposes novel efficient data-driven channel prediction algorithms that reduce pilot requirements by integrating transfer and meta-learning with a novel long-short-term decomposition (LSTD) of the linear predictors. Unlike the prior articles reviewed
above, the proposed methods apply to multi-antenna frequency-selective channels whose statistics change across frames (see Figure 1). Specific contributions are as follows.

- We develop efficient predictors for single-antenna frequency-flat channels based on transfer/meta-learned quadratic regularization. Transfer and meta-learning are used to leverage data from multiple frames in order to extract shared useful knowledge that can be used for prediction on the current frame (see Figure 2).

- Targeting multi-antenna frequency-selective channels, we introduce the LSTD-based model class of linear predictors that builds on the well-known disaggregation of standard channel models into long-term space-time signatures and fading amplitudes \[5,41–44\]. Accordingly, the channel is described by multipath features, such as angle of arrivals, delays, and path loss, that change slowly across the frame, as well as by fast-varying fading amplitudes. Transfer learning and meta-learning algorithms for LSTD-based prediction models are proposed that build on equilibrium propagation (EP) and alternating least squares (ALS).

- Numerical results under the 3GPP 5G standard channel model demonstrate the impact of transfer and meta-learning on reducing the number of pilots for channel prediction, as well as the merits of the proposed LSTD parametrization.

Part of this paper was presented in [45], which only covered meta-learning for the case of single-antenna frequency-flat channels. As compared to [45], this journal version includes both transfer and meta-learning, and it addresses the general scenario of multi-antenna frequency-selective channels by introducing and developing the LSTD model class of linear predictors.

### 1.3. Organization

The rest of the paper is organized as follows. In Section 2, we detail system and channel models, and describe conventional, transfer, and meta-learning concepts. In Section 3, we develop solutions for single-antenna frequency-flat channels. In Section 4, multi-antenna frequency-selective channels are considered, and we propose LSTD-based linear prediction schemes. Numerical results are presented in Section 5, and conclusions are presented in Section 6.

Notation: In this paper, \((\cdot)^\top\) denotes the transposition; \((\cdot)^\dagger\) the Hermitian transposition, \((\cdot)_F\) the Frobenius norm, \(|\cdot|\) the absolute value, \(||\cdot||\) the Euclidean norm, \(\text{vec}(\cdot)\) the
vectorization operator that stacks the columns of a matrix into a column vector, $[·]_i$ the $i$-th element of the vector, and $I_S$ the $S \times S$ identity matrix for some integer $S$.

2. System Model

2.1. System Model

As shown in Figure 1, we study a frame-based transmission system, with each frame containing multiple time slots. Each frame carries data from a possibly different user to the same receiver, e.g., a base station. The receiver has $N_R$ antennas, and the transmitters have $N_T$ antennas. The channel $h_{l,f}$ in slot $l = 1, 2, \ldots$ of frame $f = 1, 2, \ldots$ is a vector with $S = N_R N_T W$ entries, with $W$ being the delay spread measured in number of transmission symbols within each frame $f$, the multi-path channels $h_{l,f} \in C^{N_R N_T W \times 1}$ are characterized by fixed, frame-dependent, average path powers, path delays, Doppler spectra, and angles of arrival and departure [46]. For instance, in a frame $f$, we may have a slow-moving user in line-of-sight condition subject to time-invariant fading, whereas in another, the channel may have significant scattering with fast temporal variations with a large Doppler frequency. In both cases, the frame is assumed to be short enough that average path powers, path delays, Doppler spectra, and angles of arrival and departure do not change within the frame [41, 42].

As also seen in Figure 1, for each frame $f$, we are interested in addressing the lag-$\delta$ channel prediction problem, in which channel $h_{l+\delta,f}$ is predicted based on the $N$ past channels:

$$H_{l,f}^N = [h_{l,f}, \ldots, h_{l-N+1,f}] \in C^{S \times N}. \tag{1}$$

We adopt linear prediction with regressor $V_f \in C^{SN \times S}$, so that the prediction is given as

$$\hat{h}_{l+\delta,f} = V_f^\dagger \text{vec}(H_{l,f}^N). \tag{2}$$

The focus on linear prediction is justified by the optimality of linear estimation for Gaussian stationary processes [47], which provide standard models for fading channels in rich scattering environments.

Assuming no prior knowledge of the channel model, we adopt a data-driven approach to the design of the predictor (2). Accordingly, to train the linear predictor (2), for any frame $f$, the receiver is assumed to have available the training set

$$Z_f^{tr} = \{(x_{i,f}, y_{i,f})\}_{i=1}^{L_f} \equiv \{(\text{vec}(H_{i,f}^N), h_{l+\delta,f})\}_{i=1}^{L_f + N - 1} \tag{3}$$

encompassing $L_f$ input–output examples. Dataset $Z_f^{tr}$ can be constructed from $L_f + N + \delta - 1$ channels $\{h_{l,f}, \ldots, h_{L_f + N + \delta - 1,f}\}$ by using the lag-$\delta$ channel $h_{l+\delta,f}$ as label for the covariate vector vec$(H_{l,f}^N)$. In practice, the channel vectors $h_{l,f}$ are estimated by using pilot symbols, and estimation noise can be easily incorporated in the model (see Section 2.5). Throughout, we implicitly assume that the channels $h_{l,f}$ correspond to estimates available at the receiver.

From dataset $Z_f^{tr}$ in (3), we write the corresponding $L_f \times SN$ input matrix $X_f^{tr} = [x_{1,f}^T, \ldots, x_{L_f,f}^T]^T$, and the $L_f \times S$ target matrix $Y_f^{tr} = [y_{1,f}^T, \ldots, y_{L_f,f}^T]^T$, so that the dataset can be expressed as the pair $Z_f^{tr} = (X_f^{tr}, Y_f^{tr})$.

2.2. Channel Model

We adopt the standard spatial channel model [46]. Accordingly, a channel vector $h_{l,f}$ for slot $l$ in frame $f$, is obtained by sampling the continuous-time multipath vector channel impulse response

$$h_{l,f}(\tau) = \sum_{d=1}^{D} \sqrt{\Omega_{a,d,f}} \phi(\tau - \tau_{d,f}) \exp(-j2\pi \gamma_{d,f} \tau), \tag{4}$$
which is the sum of contributions from \( D \) paths. In (4), the waveform \( g(\tau) \) is given by the convolution of the transmitted waveform and the matched filter at the receiver. Furthermore, the contribution of the \( d \)-th path depends on the average power \( \Omega_{d,f} \), the path delay \( \tau_{d,f} \), the \( N_T N_R \times 1 \) spatial vector \( d_{a,d,f} \), the Doppler frequency \( \gamma_{d,f} \), and the starting wall-clock time of the \( l \)-th slot \( t_l \). The average power \( \Omega_{d,f} \) path delays \( \tau_{d,f} \), spatial vector \( d_{a,d,f} \), and Doppler frequency \( \gamma_{d,f} \) are constant within one frame because they depend on large-scale geometric features of the propagation environment. However, they may change over frames following Clause 7.6.3.2 (Procedure B) in [46]. The number of paths is assumed without loss of generalization to be the same for all frames \( f \) because one can set \( \Omega_{d,f} = 0 \) for frames with a smaller number of paths.

In [46], the spatial vector \( d_{a,d,f} \) has a structure that depends on field patterns and steering vectors of the transmit and receive antennas, as well on the polarization of the antennas. Mathematically, the entry of the spatial vector \( d_{a,d,f} \) corresponding to the receive and transmit antenna element \( n_R \) and \( n_T \) can be modeled as [46]

\[
[a_{d,f}]_{nR+(nT-1)NR} = \mathbf{F}_{RX,nR}(\theta_{d,f,ZOA}, \phi_{d,f,AOA})^T \mathbf{M}_{d,f} \\
\cdot \mathbf{F}_{RX,nT}(\theta_{d,f,ZOD}, \phi_{d,f,AOD}) \exp \left(-\frac{j2\pi f_{d,f,nR,nT}}{\lambda_0}\right),
\]

where \( \mathbf{F}_{RX,nR}(\cdot, \cdot) \) and \( \mathbf{F}_{RX,nT}(\cdot, \cdot) \) are the \( 2 \times 1 \) field patterns, \( \theta_{d,f,ZOA}, \phi_{d,f,AOA} \), \( \theta_{d,f,ZOD} \), and \( \phi_{d,f,AOD} \) are the zenith angle of arrival (ZOA), azimuth angle of arrival (AOA), zenith angle of departure (ZOD), and azimuth angle of departure (AOD) (in degrees), \( \lambda_0 \) is the wavelength (in m) of the carrier frequency, \( d_{a,d,f,nR,nT} \) is the length of the path (in m) between the two antennas, and \( \mathbf{M}_{d,f} \) is the polarization coupling matrix defined as

\[
\mathbf{M}_{d,f} = \begin{pmatrix}
\exp \left(j\Phi_{d,f}^{\phi\theta}\right) & \sqrt{1/\kappa_{d,f}} \exp \left(j\Phi_{d,f}^{\phi\phi}\right) & \exp \left(j\Phi_{d,f}^{\phi\theta}\right) \\
\sqrt{1/\kappa_{d,f}} \exp \left(j\Phi_{d,f}^{\phi\phi}\right) & \exp \left(j\Phi_{d,f}^{\phi\phi}\right) & \exp \left(j\Phi_{d,f}^{\phi\theta}\right) \\
\end{pmatrix},
\]

with random initial phase \( \Phi_{d,f}^{\phi\theta} \sim U(-\pi, \pi) \) and log-normal distributed cross polarization power ratio (XPR) \( \kappa_{d,f} > 0 \) [46].

In order to obtain the \( S \times 1 \) vector \( h_{l,f} \), we sample the continuous-time channel \( h_{l,f}(\tau) \) in (4) at Nyquist rate \( 1/T \) to obtain \( W \) discrete-time \( N_R N_T \times 1 \) channel impulse response

\[
h_{l,f}[w] = h_{l,f}((w-1)T)
\]

for \( w = 1, \ldots, W \). Following [41], the channel vector \( h_{l,f} \in \mathcal{C}^{N_R N_T W \times 1} \) is obtained by concatenating the \( W \) channel vectors \( h_{l,f}[w] \) for \( w = 1, \ldots, W \) as

\[
h_{l,f} = [h_{l,f}[1], \ldots, h_{l,f}[W]]^T.
\]

2.3. Conventional Learning

The optimization of the linear predictor \( V_f \) in (2) can be formulated as a supervised learning problem as it will be detailed in Section 3. In conventional learning, the predictor \( V_f \) is designed separately in each frame \( f \) based on the corresponding dataset \( \mathcal{Z}_{f}^{tr} \). In order for this predictor \( V_f \) to generalize well to slots in the same frame \( f \) outside the training set, it is necessary to have a sufficiently large number of training slots, \( L^{tr} \) [48,49].

2.4. Transfer Learning and Meta-Learning

In conventional learning, the number of required training slots \( L^{tr} \) can be reduced by selecting hyperparameters in the learning problem that reflect prior knowledge about the prediction problem at hand. In the next sections, we will explore solutions that optimize such hyperparameters based on data received from multiple previous frames. To this end, as illustrated in Figure 2, we assume the availability of channel data collected from
F frames received in the past. In each frame, the channel follows the model described in Section 2.2. Accordingly, data from previous frames consists of \( L + N + \delta - 1 \) channels \( \{ h_{1 \f, \cdots, h_{L+N+\delta-1 \f} \} \} \) for some integer \( L \).

By using these channels, the dataset

\[
Z_f = \{(x_{i \f, y_{i \f}}) \}_{i=1}^{L} \equiv \{(\text{vec}(H_{i \f}^N), h_{i+\delta \f})\}_{i=N}^{L+N-1}
\]

(9)

can be obtained as explained in Section 2.1, where \( L \) is typically larger than \( L^\f \), although this will not be assumed in the analysis. Correspondingly, we also define the \( L \times N \) input matrix \( X_f \) and the \( L \times 1 \) target vector \( y_f \). We will propose methods that leverage the historical knowledge available from dataset \( Z_f \) for \( f = 1, \ldots, F \) via transfer learning and meta-learning with the goal of reducing number of pilots, \( L^\f \), needed for channel prediction in a new frame (i.e., frame \( F + 1 \) in Figure 2).

2.5. Incorporating Estimation Noise

Until now, we assumed that channel vectors \( h_{l \f} \) are available noiselessly to the predictor. In practice, channel information needs to be estimated via pilots. To elaborate on this point, let us assume the received signal model

\[
y_{l \f}^P[i] = h_{l \f}x_{l \f}^P[i] + n_{l \f}[i],
\]

(10)

where \( x_{l \f}^P[i] \) stands for the \( i \)th transmitted pilot symbol in block \( l \) of frame \( f \), \( y_{l \f}^P[i] \) for the corresponding received signal, and \( h_{l \f} \) for the channel with additive white complex Gaussian noise \( n_{l \f}[i] \sim \mathcal{CN}(0, N_0 I_6) \). Given an average energy constraint \( E[x_{l \f}^P[i]^2] = E_x \) for the training symbol, the average signal-to-noise ratio (SNR) is given as \( E_x/N_0 \). From (10), we can estimate the channel as

\[
h_{l \f} = \frac{y_{l \f}^P[i]}{x_{l \f}^P[i]} = h_{l \f} + n_{l \f}[i] / x_{l \f}^P[i] = h_{l \f} + \xi,
\]

(11)

which suffers from channel estimation noise \( \xi \sim \mathcal{CN}(0, \text{SNR}^{-1} I_6) \). If \( P \) training symbols are available in each block, the channel estimation noise can be reduced via averaging to \( \text{SNR}^{-1}/P \). Channels \( h_{l \f} \) can be used as training data in the schemes described in the previous subsections. More efficient channel estimation methods, including sparse Bayesian learning [50] and approximate message passing approaches [51] may further reduce the channel estimation noise.

3. Single-Antenna Frequency-Flat Channels

In this section, we propose transfer learning and meta-learning methods for single-antenna flat-fading channels, which result in \( S = 1 \). Throughout this section, we write the prediction matrix \( V_f \in \mathbb{C}^{SN \times S} \) in (2) as the vector \( v_f \in \mathbb{C}^{N \times 1} \), and the target data \( Y^\f_f \in \mathbb{C}^{L^\f \times S} \) as the vector \( y^\f_f \in \mathbb{C}^{L^\f \times 1} \). Correspondingly, we rewrite the linear predictor (2) as

\[
h_{l+\delta \f} = v_f \text{vec}(H_{l \f}^N).
\]

(12)

3.1. Conventional Learning

Assuming the standard quadratic loss, we formulate the supervised learning problem as the ridge regression optimization

\[
v^* = \arg\min_{v_f \in \mathbb{C}^{N \times 1}} \left\{ ||X^\f v_f - y^\f||^2 + \lambda ||v_f - \bar{v}||^2 \right\},
\]

(13)
with hyperparameters \((\lambda, \sigma)\) given by the scalar \(\lambda > 0\) and by the \(N \times 1\) bias vector \(\sigma\). The bias vector \(\sigma\) can be thought of defining the prior mean of the predictor \(v_f\), whereas \(\lambda > 0\) specifies the precision (i.e., inverse of the variance) of this prior knowledge. The solution of problem (13) can be obtained explicitly as

\[
v^*(Z_f^T|\sigma) = (A_f^T)^{-1}((X_f^T)^ty_f + \lambda \sigma), \quad \text{with} \quad A_f^T = (X_f^T)^TX_f^T + \lambda I.
\]  

(14)

3.2. Transfer Learning

Transfer learning uses datasets \(Z_f\) in (58) from the previous \(F\) frames, i.e., with \(f = 1, \ldots, F\), to optimize the hyperparameter vector \(\sigma\) in (13) as

\[
\sigma^{\text{trans}} = \arg \min_{\sigma \in \mathbb{C}^{N \times 1}} \left\{ \sum_{f=1}^{F} \left\| X_f \sigma - y_f \right\|^2 \right\}.
\]  

(15)

The rationale for this choice is that vector \(\sigma^{\text{trans}}\) provides a useful prior mean to be used in the ridge regression problem (13), because it corresponds to an optimized predictor for the previous frames. Having optimized the bias vector \(\sigma^{\text{trans}}\), we train a channel predictor \(v\) via ridge regression (13) by using the training data \(Z_{f_{\text{new}}}\) for a new frame \(f_{\text{new}}\) with \(L_{\text{new}}\) training samples, to obtain

\[
v^*_{f_{\text{new}}} = v^*(Z_{f_{\text{new}}}|\sigma^{\text{trans}}).
\]  

(16)

Note that during deployment time, this approach has the same computational complexity as that of conventional learning, because the bias vector is treated as a constant vector.

3.3. Meta-Learning

Unlike transfer learning, which utilizes all the available datasets \(\{Z_f\}_{f=1}^{F}\) from the previous frames at once as in (15), meta-learning allows for the separate adaptation of the predictor in each frame. To this end, for each frame \(f\), we split the \(L\) data points into \(L^{tr}\) training pairs \(\{(x_{i,f}, y_{i,f})\}_{i=1}^{L^{tr}}\) and \(L^{te}\) test pairs \(\{(x_{i,f}, y_{i,f})\}_{i=1}^{L^{te}}\), resulting in two separate datasets, \(Z_f^{tr}\) and \(Z_f^{te}\). We correspondingly define the \(L^{tr} \times N\) input matrix \(X_f^{tr}\) and the \(L^{tr} \times 1\) target vector \(y_f^{tr}\), as well as the \(L^{te} \times N\) input matrix \(X_f^{te}\) and the \(L^{te} \times 1\) target vector \(y_f^{te}\).

The hyperparameter vector \(\sigma\) is then optimized by minimizing the sum of the loss of the predictors \(v^*(Z_f^{tr}|\sigma)\) in (13) that are adapted separately for each frame \(f = 1, \ldots, F\) given the bias vector \(\tilde{\sigma}\). Accordingly, estimating the loss in each frame \(f\) via the test set \(Z_f^{te}\) yields the meta-learning problem

\[
\sigma^{\text{meta}} = \arg \min_{\sigma \in \mathbb{C}^{N \times 1}} \left\{ \sum_{f=1}^{F} \left\| v^*(Z_f^{tr}|\sigma)^T x_{i,f}^{te} - y_{i,f}^{te} \right\|^2 \right\}.
\]  

(17)

As studied in [52], the minimization in (17) is a least squares problem that can be solved in closed form as

\[
\sigma^{\text{meta}} = \arg \min_{\sigma \in \mathbb{C}^{N \times 1}} \sum_{f=1}^{F} \left\| X_f^{te} \sigma - y_f^{te} \right\|^2 = (X^T X)^{-1} X^T \tilde{y},
\]

(18)

where \(L^{te} \times N\) matrix \(X_f^{te}\) contains by row the Hermitian transpose of the \(N \times 1\) preconditioned input vectors \(\{\Lambda(A_f^T)^{-1} x_{i,f}^{te}\}_{i=1}^{L^{te}}\), with \(A_f^{tr} = (X_f^{tr})^TX_f^{tr} + \lambda I\); \(y_f^{te}\) is \(L^{te} \times 1\) vector containing vertically the complex conjugate of the transformed outputs \(\{(y_{i,f}^{te})^T X_f^{tr}(A_f^{tr})^{-1} x_{i,f}^{te}\}_{i=1}^{L^{te}}\).
\[ \mathbf{\tilde{X}}_{\text{te}} \approx \mathbf{g} \]

and the FL \( F \) \( \times \) \( 1 \) vector \( \mathbf{\tilde{y}} = [\mathbf{\tilde{y}}_{\text{te}}^1, \ldots, \mathbf{\tilde{y}}_{\text{te}}^F]^\top \) stacks vertically the \( L_{\text{te}} \) \( \times \) \( 1 \) vectors \( \{\mathbf{\tilde{y}}_{\text{te}}^f\}_{f=1}^F \). Unlike standard meta-learning algorithms used by most papers on communications [25,29,30,32–34], the proposed meta-learning procedure adopts linear models, significantly reducing the computational complexity of meta-learning [52].

After meta-learning, similar to transfer learning, based on the meta-learned hyperparameter \( \mathbf{\bar{V}}_{\text{meta}} \), we train a channel predictor via ridge regression (13), obtaining

\[
V_{\text{new}}^* = V^* (Z_{\text{new}} | V_{\text{meta}}).
\]  

4. Multi-Antenna Frequency-Selective Channels

In this section, we study the more general scenario with any number of antennas and with frequency-selective channels, resulting in \( S > 1 \). As we will discuss, a naïve extension of the techniques presented in the previous sections is undesirable, because this would not leverage the structure of the channel model (4). For this reason, in the following, we will introduce novel hybrid model- and data-driven solutions that build on the channel model (4).

4.1. Naïve Extension

We start by briefly presenting the direct extension of the approaches studied in the previous section to any \( S > 1 \). Unlike the previous section, we adopt the general matrix notation introduced in Section 2. First, with \( S = 1 \), conventional learning obtains the predictor by solving problem (13), which is generalized to any \( S > 1 \) as the minimization

\[
V^* (Z_{\text{tr}}^f | \mathbf{\bar{V}}) = \arg\min_{V_f \in C^{SN \times S}} \left\{ ||X_{\text{tr}}^f V_f - Y_{\text{tr}}^f||_F^2 + \lambda ||V_f - \mathbf{\bar{V}}||_F^2 \right\},
\]  

over the linear prediction matrix \( V_f \) in (2). Similarly, transfer learning computes the bias matrix \( \mathbf{\bar{V}}_{\text{trans}} \) by solving the following generalization of problem (15),

\[
\mathbf{\bar{V}}_{\text{trans}} = \arg\min_{V \in C^{SN \times S}} \left\{ \sum_{f=1}^F \left( ||X_{\text{tr}}^f V - Y_{\text{tr}}^f||_F^2 \right) \right\},
\]  

followed by the evaluation of the predictor \( V^* (Z_{\text{tr}}^f | \mathbf{\bar{V}}_{\text{trans}}) \) using (20); whereas meta-learning addresses the following generalization of minimization (17),

\[
\mathbf{\bar{V}}_{\text{meta}} = \arg\min_{V \in C^{SN \times S}} \left\{ \sum_{f=1}^F \sum_{i=1}^{L_{\text{te}}} ||V^* (Z_{\text{tr}}^f | \mathbf{\bar{V}} + \mathbf{\bar{V}}_{\text{meta}}) \mathbf{x}_{i,f} - y_{i,f}||_2^2 \right\},
\]  

over the bias matrix \( \mathbf{\bar{V}} \in C^{SN \times S} \), which is used to compute the predictor \( V^* (Z_{\text{tr}}^f | \mathbf{\bar{V}}_{\text{meta}}) \) in (20).

The issue with the naïve extensions (21) and (22) is that the dimension of the predictor \( V \) and of the hyperparameter matrix \( \mathbf{\bar{V}} \) can become extremely large when \( S \) grows. This, in turn, may lead to overfitting in the hyperparameter space [53] when the number of frames, \( F \), is limited. This form of overfitting may prevent transfer learning and meta-learning from effectively reducing the sample complexity for problem (20), because the optimized hyperparameter matrix \( \mathbf{\bar{V}} \) would be excessively dependent on the data received in the \( F \) previous frames. To solve this problem, we propose next to utilize the structure of the channel model (4) in order to reduce the dimension of the channel parametrization.

4.2. LSTD Channel Model

The channel model (4) implies that the channel vector \( h_{i,f} \) in (7) and (8) can be written as the product of a frame-dependent \( N_R N_T W \times D \) matrix \( T_f \) and of a slot-dependent \( D \times 1 \) vector \( \beta_{i,f} \) as in [41],

\[ \mathbf{\tilde{X}}_{\text{te}} \approx \mathbf{g} \]
where $T_f$ collects space-time signatures of the $D$ paths as

$$T_f = [\Omega_{1/f}^{1/2} \mathbf{g}(\tau_{1,f}) \otimes \text{vec}(a_{1,f}), \ldots, \Omega_{D/f}^{1/2} \mathbf{g}(\tau_{D,f}) \otimes \text{vec}(a_{D,f})],$$

with $\mathbf{g}(\tau_{l,f}) = [\mathbf{g}(-\tau_{l,f}), \ldots, \mathbf{g}((W-1)T-\tau_{l,f})]^\top$ being the $W \times 1$ vector that collects the Nyquist-rate samples of the delayed waveform $g(\tau - \tau_{l,f})$, and the $D \times 1$ fading amplitude vector being defined as $\beta_{l,f} = [\exp(-j\omega_{1,f}t_l), \ldots, \exp(-j\omega_{D,f}t_l)]^\top$.

The frame-dependent matrix $T_f$ is typically rank-deficient, because paths are generally not all resolvable [54,55]. To account for this structural property of the channel, as in [41], we introduce a $N_RW \times K$ full-rank unitary matrix $B_f$, such that $\text{span}\{T_f\} = \text{span}\{B_f\}$ and redefine (23) as

$$h_{l,f} = B_fd_{l,f}. \quad (25)$$

As an example, the unitary matrix $B_f$ can be obtained from the singular value decomposition of matrix $T_f$, i.e., $T_f = B_f \Lambda_f^{1/2} U_f$, by introducing the $K \times 1$ vector $d_{l,f} = \Lambda_f^{1/2} U_f \beta_{l,f} [41]$. For future reference, we also rewrite (25) as

$$h_{l,f} = \sum_{k=1}^{K} b_k^f d_{l,f}^k, \quad (26)$$

where $d_{l,f}^k$ is the $k$-th element of the vector $d_{l,f}$ and $b_k^f$ is the $k$-th column of the matrix $B_f$.

We will refer to matrix $B_f$ in (26) as the long-term space-time feature matrix, or feature matrix for short, whereas vector $d_{l,f}$ will be referred as the short-term corresponding amplitude vector. Parametrization (25) and (26) are particularly efficient when the feature matrix $B_f$ can be accurately estimated from the available data. For conventional learning, this requires observing a sufficiently large number of slots per frame, i.e., a large $L_{\text{new}}$ [41], as well as a channel that varies sufficiently quickly across each frame. In contrast, as we will explore, transfer and meta-learning can potentially leverage data from multiple frames in order to enhance the estimation of the feature matrix.

### 4.3. LSTD-Based Prediction Model

Given the LSTD channel model (25) and (26), in this subsection we redefine the problem of predicting channel $h_{l+\delta,f} = B_fd_{l+\delta,f}$ as the problem of estimating the feature matrix $B_f$ and predicting the amplitude vector $d_{l+\delta,f}$ based on the available data. This will lead to a reduced-rank parametrization of the linear predictor (2).

To start, we write the predicted channel $\hat{h}_{l+\delta,f}$ as

$$\hat{h}_{l+\delta,f} = \hat{B}_f \hat{d}_{l+\delta,f}, \quad (27)$$

where $\hat{B}_f$ and $\hat{d}_{l+\delta,f}$ are the estimated feature matrix and the predicted amplitude vector, respectively. To define the corresponding predictor, we first observe that the input matrix $H_{l,f}^N$ in (1) can be expressed by using (25) as

$$H_{l,f}^N = B_f [d_{l,f}, \ldots, d_{l-N+1,f}], \quad (28)$$

Assume now that we have an estimated feature matrix $\hat{B}_f$. If this estimate is sufficiently accurate, the $N$ past amplitudes $[d_{l,f}, \ldots, d_{l-N+1,f}] \in \mathbb{C}^{K \times N}$ can be in turn estimated from $H_{l,f}^N$ as

$$[\hat{d}_{l,f}, \ldots, \hat{d}_{l-N+1,f}] = \hat{B}_f^\dagger H_{l,f}^N. \quad (29)$$
Consider now the prediction of the $k$-th amplitude $d_{l+\delta,f}^k$. Generalizing (12), we adopt the linear predictor
\[
\hat{d}_{l+\delta,f}^k = (v_f^k)^\dagger \text{vec}([d_{l,f}^k, \ldots, d_{l-N+1,f}^k]),
\]
(30)
where $v_f^k$ is an $N \times 1$ prediction vector, and
\[
[d_{l,f}^k, \ldots, d_{l-N+1,f}^k] = (\hat{b}_f^k)^\dagger H_{l,f} \in \mathbb{C}^{1 \times N}
\]
is the $k$-th row of the matrix (29), which represents the past $N$ fading scalar amplitudes that correspond to the $k$-th feature $b_f^k$. Plugging the prediction (30) into (27) yields the predicted channel $\hat{h}_{l+\delta,f}$ (cf. (26))
\[
\hat{h}_{l+\delta,f} = \sum_{k=1}^{K} \hat{b}_f^k \hat{d}_{l+\delta,f}^k.
\]
(32)

As detailed in Appendix A, inserting (30) and (31) to (32), we can express the LSTD-based prediction (32) in the form (2) as
\[
\hat{h}_{l+\delta,f} = (V_f^{(K)})^\dagger \text{vec}(H_{l,f}^N),
\]
(33)
where the LSTD-based predictor matrix $V_f^{(K)} \in \mathbb{C}^{SN \times S}$ is given as
\[
V_f^{(K)} = \sum_{k=1}^{K} v_f^k \otimes (\hat{b}_f^k(\hat{b}_f^k)^\dagger),
\]
(34)
where $\otimes$ is the Kronecker product. Overall illustration of LSTD-based channel prediction is summarized in Figure 3.

![Figure 3](image-url)

**Figure 3.** Illustration of the considered LSTD-based prediction model. (i) Estimate amplitudes $d_{l,f}$ via the estimated long-term feature matrix $\hat{B}_f$. (ii) Feature-wise short-term fading amplitude prediction $\hat{d}_{l+\delta,f}^k$ based on feature-wise predictor $v_f^k$ for $k = 1, \ldots, K$. (iii) Reconstruction of the predicted channel $\hat{h}_{l+\delta,f}$ and predicted fading amplitude $\hat{d}_{l+\delta,f}$.

The LSTD prediction model (33) reduces the dimension of the learnable parameters from $S^2N$ (for $V_f$) to $(S + N)K$ (for $V_f^{(K)}$). This complexity reduction comes at a minimal cost in terms of bias as long as the number of total features $K$ is accurately chosen (a detailed discussion can be found in Section 4.7) and correlations across amplitudes $d_{l,f}^k$ for different features $k = 1, \ldots, K$ are negligible.
4.4. Conventional Learning for LSTD-Based Prediction

In conventional learning, the goal is to optimize the LSTD-based predictor $V_f^{(K)}$ by optimizing the feature matrix $\hat{B}_f$ and the feature-wise predictors $\{v_f^k\}_{k=1}^K$ based on the available training dataset $Z_f^{tr}$. Substituting $V_f$ with $V_f^{(K)}$ defined in (34) into the naïve extension of conventional learning in (20) yields the problem

$$V_f^{(K),*}(Z_f^{tr} | V_f^{(K)}) = \arg \min_{\hat{B}_f,v_f^1,...,v_f^K} ||X_f^{tr}V_f^{(K)} - Y_f^{tr}||_F^2 + \lambda ||V_f - V_f^{(K)}||_F^2,$$

subject to $\hat{B}_f^T\hat{B}_f = I_K,$ (35)

over the optimization variables $(\hat{B}_f, \{v_f^k\}_{k=1}^K)$. In (35), the hyperparameters $(\lambda, V_f^{(K)})$ are given by the scalar $\lambda > 0$ and by the $S \times S$ LSTD-based bias matrix $V_f^{(K)}$ defined as (cf. (34))

$$V_f^{(K)} = \sum_{k=1}^K \sigma^k \otimes (\hat{b}_k^k)^{\dagger}. \quad (36)$$

Because the Euclidean norm regularization $||V_f - V_f^{(K)}||_F^2$ in (35) mixes long-term and short-term dependencies due to (34) and (36), we propose the modification of problem (35)

$$V_f^{(K),*}(Z_f^{tr} | (\hat{b}_f^k, \sigma^k)_{k=1}^K) = \arg \min_{\hat{B}_f,v_f^1,...,v_f^K} \left\{ ||X_f^{tr}V_f^{(K)} - Y_f^{tr}||_F^2 + \lambda_2 \sum_{k=1}^K ||v_f^k - \sigma^k||^2 \right\},$$

subject to $\hat{B}_f^T\hat{B}_f = I_K,$ (37)

with hyperparameters $(\lambda_1, \lambda_2, \hat{b}_f^1, \ldots, \hat{b}_f^K, \sigma^1, \ldots, \sigma^K)$ given by the scalars $\lambda_1, \lambda_2 > 0$, by the $S \times 1$ long-term bias vectors $\hat{b}_f^1, \ldots, \hat{b}_f^K$, and by the $N \times 1$ short-term bias vectors $\sigma^1, \ldots, \sigma^K$.

For each feature $k$, the considered regularization minimizes the Euclidean distance between the short-term prediction vector $v_f^k$ and the short-term bias vector $\sigma^k$ as in Section 3, while maximizing the alignment between the long-term feature vector $\hat{b}_f^k$ and the long-term bias vector $\hat{b}_f^k$ in a manner akin to the kernel alignment method of [56].

To address problem (37), inspired by [57,58], we propose a sequential approach, in which the pair $(v_f^k, \hat{b}_f^k)$ consisting of the $k$-th predictor $v_f^k$ and the $k$-th feature vector $\hat{b}_f^k$ is optimized in the order $k = 1, 2, \ldots, K$. Specifically, at each step $k$, we consider the problem

$$\hat{b}_f^{k,*}, v_f^{k,*} = \arg \min_{\hat{b}_f^{k}, v_f^{k}} \left\{ ||X_f^{tr}(V_f^{(K)})^k - (Y_f^{tr})^k||_F^2 - \lambda_1 \text{tr}\left( (\hat{b}_f^k)^\dagger(\hat{b}_f^k)^\dagger\hat{b}_f^k \right) + \lambda_2 ||v_f^k - \sigma^k||^2 \right\},$$

subject to $(\hat{b}_f^k)^\dagger\hat{b}_f^k = 1,$ (38)

where the $L^{tr} \times S$ $k$-th residual target matrix $(Y_f^{tr})^k$ is defined as [57,58]

$$(Y_f^{tr})^k = \begin{cases} Y_f^{tr}, & \text{for } k = 1, \\ Y_f^{tr} - \sum_{k=1}^{k-1} X_f^{tr}(V_f^{(K)})^{k,*}, & \text{for } k > 1, \end{cases} \quad (39)$$
given the $k$-th predictor

$$
(V_f^{(K)})^k = v_f^k \otimes (\hat{b}_f^k(\hat{b}_f^k)^*)
$$

and $k$-th optimized predictor

$$
(V_f^{(K)})^{k,*} = v_f^{k,*} \otimes (\hat{b}_f^{k,*}(\hat{b}_f^{k,*})^*)
$$

Because (38) is a nonconvex problem, we consider alternating least squares (ALS) [59] to obtain the optimal solution $\{\hat{b}_f^{k,*}, v_f^{k,*}\}$ by iterating between the following steps: (i) for a fixed $\hat{b}_f^k$, update $v_f^k$ as

$$
v_f^k \leftarrow \arg \min_{v_f^k} \left\{ \|X_f^*(V_f^{(K)})^k - (Y_f^*)^k\|^2_F + \lambda_2 \|v_f^k - \sigma^k\|^2 \right\};
$$

(ii) for a fixed $v_f^k$, update $\hat{b}_f^k$ as

$$
\hat{b}_f^k \leftarrow \arg \min_{\hat{b}_f^k} \left\{ \|X_f^*(V_f^{(K)})^k - (Y_f^*)^k\|^2_F - \lambda_1 \text{tr} \left( (\hat{b}_f^k)^*(\hat{b}_f^k)^*) \right) \right\},
$$

subject to $(\hat{b}_f^k)^*(\hat{b}_f^k) = 1$.

until convergence. Closed-form solutions for (42) and (43) can be found in Appendix B, and the overall LSTD-based conventional learning scheme can be found in Algorithm 1.

**Algorithm 1: LSTD-based conventional learning for channel prediction for $S \geq 1$**

**Input:** training data set $Z_f^*$ at current frame $f$; number of total features $K$, hyperparameter $\lambda_1 > 0, \lambda_2 > 0, \hat{b}_1, \ldots, \hat{b}_K, \sigma_1, \ldots, \sigma_K$

**Output:** learned LSTD-based predictor $V_f^{(K),*}$

**LSTD–based conventional learning:**

initialize residual target matrix $(Y_f^*)^1 \leftarrow Y_f^*$

for each feature $k = 1 \ldots, K$ do

$v_f^{k,*}, \hat{b}_f^{k,*} = \text{ALS}(k, X_f^*, (Y_f^*)^k, \lambda_1, \lambda_2, \hat{b}_k, \sigma_k)$

update residual target matrix $(Y_f^*)^{k+1} \leftarrow (Y_f^*)^k - X_f^* v_f^k \otimes (\hat{b}_f^k(\hat{b}_f^k)^*)$

end

return $V_f^{(K),*} = \sum_{k=1}^K v_f^k \otimes (\hat{b}_f^k(\hat{b}_f^k)^*)$

while not converged do

first update $v_f^k$ via (42) then, update $\hat{b}_f^k$ via (43)

end

return $v_f^k, \hat{b}_f^k$
4.5. Transfer Learning for LSTD-Based Prediction

Similar to conventional learning, transfer learning for LSTD-based prediction can be addressed from the naïve extension (21) by utilizing the LSTD parametrization \( V^{(K)} \) in (34) in lieu of the unconstrained predictor \( V \) to obtain the bias matrix \( \bar{V}^{(K)\text{, trans}} \),

\[
\bar{V}^{(K)\text{, trans}} = \arg \min_{\hat{b}^{1\text{, trans}}, \ldots, \hat{b}^{K\text{, trans}}, \hat{\sigma}^{1\text{, trans}}, \ldots, \hat{\sigma}^{K\text{, trans}}} \left\{ \sum_{f=1}^{F} \|X_f V^{(K)} - Y_f\|_F^2 \right\}, \quad (Y_f)^k = \begin{cases} Y_{f, trans}, & \text{for } k = 1, \\ Y_f - \sum_{k'=1}^{k-1} X_f (V^{(K)})^{k', \text{trans}}, & \text{for } k > 1 \end{cases}
\]

which can also be solved via the ALS-based sequential approach detailed in Section 4.4. This produces the sequences \( \bar{b}^{1\text{, trans}}, \ldots, \bar{b}^{K\text{, trans}}, \hat{\sigma}^{1\text{, trans}}, \ldots, \hat{\sigma}^{K\text{, trans}} \) as (cf. (38))

\[
\bar{b}^{k\text{, trans}}, \hat{\sigma}^{k\text{, trans}} = \arg \min_{\hat{b}^{k\text{, trans}}, \hat{\sigma}^{k\text{, trans}}} \left\{ \sum_{f=1}^{F} \|X_f (V^{(K)})^k - (Y_f)^k\|_F^2 \right\}, \quad (V^{(K)})^{k, \text{trans}} = \arg \min_{\hat{b}^{k\text{, trans}}} \left\{ \sum_{f=1}^{F} \|X_f (V^{(K)})^k - (Y_f)^k\|_F^2 \right\}
\]

where the residual target matrix \( (Y_f)^k \) is defined as (cf. (39))

\[
(V^{(K)})^{k, \text{trans}} = \hat{\sigma}^{k, \text{trans}} \otimes (\bar{b}^{k, \text{trans}} (\bar{b}^{k, \text{trans}})^\dagger).
\]

Details for transfer learning can be found in Appendix C, and the overall transfer learning scheme for LSTD prediction is summarized in Algorithm 2. After transfer learning, similar to Section 3.2, based on the optimized hyperparameters \( \hat{b}^{1\text{, trans}}, \ldots, \hat{b}^{K\text{, trans}}, \hat{\sigma}^{1\text{, trans}}, \ldots, \hat{\sigma}^{K\text{, trans}} \), the LSTD-based channel predictor for a new frame \( f_{\text{new}} \) can be obtained via (37) as

\[
V^{(K)\text{, new}} = V^{(K)\text{, new}} (Z^{f_{\text{new}}\text{, trans}} \hat{b}^{1\text{, trans}}, \hat{\sigma}^{1\text{, trans}})_{k=1}^{K},
\]

which can also be solved in the sequential way as in (38).

**Algorithm 2:** LSTD-based transfer-learning for channel prediction for \( S \geq 1 \)

**Input:** previous data \( \{ \{ l, f \}_{l=1}^{L+N+\delta-1} \}_{f=1}^{F} \), window size \( N \); prediction lag size \( \delta \); number of total features \( K \)

**Output:** transfer-learned hyperparameters \( \hat{b}^{1\text{, trans}}, \ldots, \hat{b}^{K\text{, trans}}, \hat{\sigma}^{1\text{, trans}}, \ldots, \hat{\sigma}^{K\text{, trans}} \)

initialize \( X^{\text{trans}} \) and \( Y^{\text{trans}} \) to empty matrices

for each frame \( f \in \{ 1, \ldots, F \} \) do

\[
X^{\text{trans}} = [X^{\text{trans}}, X_f]^\top \text{ and } Y^{\text{trans}} = [Y^{\text{trans}}, Y_f]^\top
\]

end

follow LSTD-based conventional learning in Algorithm 1 with input

\[
Z^f = (X^{\text{trans}}, Y^{\text{trans}}), \lambda_1 = 0, \lambda_2 = 0
\]
4.6. Meta-Learning for LSTD-Based Prediction

Plugging (37) into the naïve extension of (22), we can formulate the meta-learning problem for LSTD-based prediction as

\[
\min_{\{\hat{b}^k, \hat{a}^k\}_{k=1}^F} \sum_{f=1}^F \left\| X_f^b V_f^{(K),*} \left( Z_f^r \| \{ \hat{b}_f^k, \hat{a}_f^k \}_{k=1}^K \right) - Y_f^b \right\|^2_F
\]

(49)

Similar to the sequential approach (38) described in Section 4.4, we propose a hierarchical sequential approach for meta-learning by using (38) in the order \( k = 1, \ldots, K \), obtaining the problem

\[
\hat{b}_{k, \text{meta}}, \hat{a}_{k, \text{meta}} = \arg \min_{\hat{b}_k, \hat{a}_k} \left\{ \sum_{f=1}^F \left\| X_f^b (V_f^{(K)k,*}) - (Y_f^b)k \right\|^2_F \right\}
\]

(50)

with the residual target matrix \((Y_f^b)k\) defined as (cf. (39))

\[
(Y_f^b)k = \begin{cases} 
Y_f^b, & \text{for } k = 1, \\
Y_f^b - \sum_{k=1}^{k-1} X_f^b (V_f^{(K)k,*}), & \text{for } k > 1.
\end{cases}
\]

(51)

The bilevel non-convex optimization problem (50) is addressed through gradient-based updates with gradients computed via equilibrium propagation (EP) [60,61]. EP uses finite differentiation to approximate the gradient of the bilevel optimization (50), where the difference is computed between two gradients obtained at two stationary points \((\hat{b}_f^{k,*}, \hat{v}_f^{k,*})\) and \((\hat{b}_f^{k,x}, \hat{v}_f^{k,x})\) for the original problem (38) and modified version of (38) that considers additional prediction loss for the test set \(Z_f^v\). Specifically, EP leverages the asymptotic equality [60]

\[
\nabla_{\hat{b}_k} \sum_{f=1}^F \left\| X_f^b (V_f^{(K)k,*}) - (Y_f^b)k \right\|^2_F = \lim_{\alpha \to 0} \frac{2\lambda_1}{\alpha} \sum_{f=1}^F \left( \hat{b}_{f,x}^{k,*} (\hat{b}_f^{k,*})^\dagger - \hat{b}_{f,x}^{k,a} (\hat{b}_f^{k,a})^\dagger \right) \hat{b}_f^k
\]

(52)

and

\[
\nabla_{\hat{a}_k} \sum_{f=1}^F \left\| X_f^b (V_f^{(K)k,*}) - (Y_f^b)k \right\|^2_F = \lim_{\alpha \to 0} \frac{2\lambda_2}{\alpha} \sum_{f=1}^F (\hat{v}_{f,x}^{k,*} - \hat{v}_{f,x}^{k,a}),
\]

(53)

with additional real-valued hyperparameter \(\alpha \in \mathbb{R}\), which is generally chosen to be a non-zero small value [60,61]. In (52) and (53), vectors \(\hat{b}_f^{k,x}\) and \(\hat{v}_f^{k,x}\) are defined as (cf. (38))

\[
\hat{b}_f^{k,x}, \hat{v}_f^{k,x} = \arg \min_{\hat{b}_f^{k,x}, \hat{v}_f^{k,x}} \left\{ \left\| X_f^b (V_f^{(K)k}) - (Y_f^b)k \right\|^2_F + \alpha \left\| X_f^b (V_f^{(K)k}) - (Y_f^b)k \right\|^2_F \right\}
\]

subject to \((\hat{b}_f^{k,x})^\dagger \hat{b}_f^{k} = 1\).

(54)

Derivations for the gradients (52) and (53) can be found in Appendix D.

To reduce the computational complexity for the gradient-based updates, we adopt stochastic gradient descent with the Adam optimizer as done in [61] in order to update \(\hat{b}_k\) and \(\hat{a}_k\) based on (52) and (53). The overall LSTD-based meta-learning scheme is detailed in Algorithm 3.
After meta-learning, as in Section 3.3, based on the optimized $\theta^1_{1,\text{meta}}, \ldots, \theta^K_{1,\text{meta}}$ and $\phi^1_{1,\text{meta}}, \ldots, \phi^K_{1,\text{meta}}$, LSTD-based channel predictor for a new frame $f^\text{new}$ can be obtained via (37) as

$$V^{(K),c}_{f^\text{new}} = V^{(K),c}(Z^\text{new}_{f^\text{new}} | \{\theta^{k,\text{meta}}, \phi^{k,\text{meta}}\}_{k=1}^{K}),$$

which can be solved in a sequential way, as in (38).

The computational complexity order of the considered schemes is summarized in Tables 1 and 2. At deployment time, as seen in Table 1, all schemes require the same computational complexity of conventional learning. In contrast, in the offline meta-learning or transfer learning phase, the computational overhead depends on the dimension of the channel vector $S = N_R N_f W$. LSTD-based schemes can reduce the computational overhead as compared to naïve solutions when the channel vector is large, i.e., $S \gg 1$, and the rank $K$ is sufficiently small. This is quantified in Table 2, where $I_{\text{ALS}}$ is the number of iterations for ALS and $I_{\text{EP}}$ is the number of iterations for EP.

**Table 1.** Computational complexity analysis at deployment (meta-testing).

| Learning Type         | $O(\cdot)$ for Naïve Approach | $O(\cdot)$ for LSTD-Based Approach |
|-----------------------|--------------------------------|-----------------------------------|
| Conventional learning | $O(S^3 N^3 + S^2 N^2 L^r)$   | $O(K I_{ALS}(S^3 N^2 + S^2) + N^3 + S^3)$ |
| Transfer learning     | $O(S^3 N^3 + S^2 N^2 L^r)$   | $O(K I_{ALS}(S^3 N^2 + S^2) + N^3 + S^3)$ |
| Meta-learning         | $O(S^3 N^3 + S^2 N^2 L^r)$   | $O(K I_{ALS}(S^3 N^2 + S^2) + N^3 + S^3)$ |

**Algorithm 3:** LSTD-based meta-learning for channel prediction for $S \geq 1$

**Input:** previous data $\{\{h_{i,f}^{l=1+N_j-1}\}_{f=1}^{F}\}$; training set size $L^r$; test set size $L^t = L - L^r$; window size $N$; prediction lag size $\delta$; number of total features $K$; step size for gradient-based update $\kappa$; hyperparameter $l_1 > 0, l_2 > 0, \alpha < 1$

**Output:** meta-learned hyperparameters $\theta^{1,\text{meta}}, \ldots, \theta^{K,\text{meta}}, \phi^{1,\text{meta}}, \ldots, \phi^{K,\text{meta}}$

initialize hyperparameters $\theta^1, \ldots, \theta^K, \phi^1, \ldots, \phi^K$

for each frame $f \in \{1, \ldots, F\}$ do

split $Z_f$ into training set $Z_f^t$ and test set $Z_f^t$

initialize residual target matrices $(Y_f^t)^{k=1} \leftarrow Y_f^t, (Y_f^t)^{k=1} \leftarrow Y_f^t$

end

for each feature $k = 1, \ldots, K$ do

EP-based meta-learning for feature $k$

while not converged do

for each frame $f \in \{1, \ldots, F\}$ do

$\hat{\theta}^k_f, \hat{\phi}^k_f = \text{ALS}(k, X^t_f, (Y^t_f)^k, \lambda_1, \lambda_2, \hat{b}^k, \hat{\phi}^k)$

$\hat{\theta}^k_f, \hat{\phi}^k_a = \text{ALS}(k, [X^a_f, \sqrt{\hat{X}^a_f}]^\top, [(Y^a_f)^k, \sqrt{\hat{X}^a_f}]^\top, \lambda_1, \lambda_2, \hat{b}^k, \hat{\phi}^k)$

end

$\theta^k \leftarrow \hat{\theta}^k - \kappa \frac{2l_1}{\pi} \sum_{f=1}^{F} (\hat{\theta}^k_f - \hat{\theta}^k_a)$

$\phi^k \leftarrow \phi^k - \kappa \frac{2l_1}{\pi} \sum_{f=1}^{F} (\hat{\theta}^k_f (\hat{\phi}^k_f)^\top - \hat{\phi}^k_a (\hat{\phi}^k_a)^\top) \hat{b}^k$

end

$\theta^{k,\text{meta}} \leftarrow \theta^k, \phi^{k,\text{meta}} \leftarrow \phi^k$

update residual target matrix for both training and test set

$(Y_f)^{k+1} \leftarrow (Y_f)^k - X_f \hat{\phi}^k_f \otimes (\hat{\theta}^k_f (\hat{\phi}^k_f)^\top)$ for every frame $f$

end
Table 2. Computational complexity analysis during meta-training.

| Learning Type            | $O(\cdot)$ for Naïve Approach | $O(\cdot)$ for LSTD-Based Approach |
|--------------------------|--------------------------------|-----------------------------------|
| Conventional learning    | $O(S^3N^3 + S^2N^2L)$         |                                    |
| Transfer learning        | $O(KI_{I_{LS}}(FL(SN^2 + S^2) + N^3 + S^3))$ |                                    |
| Meta-learning            | $O(F(S^3N^3 + S^2N^2L)$ | $O(KF_{I_{I_{LS}}}(L(SN^2 + S^3) + N^3 + S^3))$ |

4.7. Rank-Estimation for LSTD-Based Prediction

The number of total features $K$ for LSTD-based predictions depends on the rank of the unknown space-time signature matrix $T_f$ as discussed in Section 4.2. This rank can be estimated by using available channels from previous frames if we assume that the number of total features does not change over multiple frames. This can be achieved via one of the standard methods, Akaike’s information theoretic criterion (AIC) (Equation (16) in [62]), which is applicable for all the proposed LSTD-based techniques. However, as the AIC-based rank estimation generally tends to be overestimated [62,63], we propose a potentially more effective estimator for meta-learning, which utilizes a validation dataset.

To this end, we first split the available $F$ frames into $F_{\text{tr}}$ meta-training frames $f = 1, \ldots, F_{\text{tr}}$ and $F_{\text{val}}$ meta-validation frames $f = F_{\text{tr}} + 1, \ldots, F$. Then, we compute the sum-loss as (cf. (49))

$$
\sum_{f=F_{\text{tr}}+1}^{F_{\text{tr}}+F_{\text{val}}} \| Y_f X_f V(k)_f \|_2^2 (Z_f^\text{tr} (b^{k,\text{meta}}, \sigma^{k,\text{meta}})_{k=1}^L) - Y_f \|_f^2
$$

(56)

where the hyperparameters $\{b^{k,\text{meta}}, \sigma^{k,\text{meta}}\}_{k=1}^L$ are computed by using the $F_{\text{tr}}$ meta-training frames, as explained in the previous section. The rank-estimation procedure sequentially evaluates the meta-validation loss (56) in order to minimize it over the selection of $k$. In this regard, it is worth noting that an increase in the total number of features always decreases the meta-training loss in (49), whereas this is not necessarily true for the meta-validation and meta-test losses.

5. Experiments

In this section, we present experimental results for the prediction of multi-antenna and/or frequency-selective channels. Numerical examples for single-antenna frequency-flat channels for both offline and online learning scenarios can be found in the conference version of this paper [45]. For all the experiments, we compute the normalized mean squared error (NMSE) $\|\hat{h}_{L+\delta,f} - h_{L+\delta,f}\|^2 / \|h_{L+\delta,f}\|^2$, which is averaged over 100 samples for 200 new frames. To avoid discrepancies between the evaluation measures used during the training and testing phase, we also adopt the NMSE as the training loss function by normalizing the training dataset for the new frame $f_{\text{new}}$ as (cf. (3))

$$
Z_{f_{\text{new}}} = \left\{ (x_{i,f_{\text{new}}}, y_{i,f_{\text{new}}}) \right\}_{i=1}^{L_{f_{\text{new}}}} \equiv \left\{ \left( \text{vec}(H_{i,f_{\text{new}}}^N) / \|h_{L+\delta,f_{\text{new}}}\|, h_{L+\delta,f_{\text{new}}}, \|h_{L+\delta,f_{\text{new}}}\| \right) \right\}_{i=1}^{L_{f_{\text{new}}} + N - 1},
$$

(57)

and similarly redefine the datasets from previous frames $f = 1, \ldots, F$ for transfer and meta-training as (cf. (58))

$$
Z_f = \left\{ (x_{i,f}, y_{i,f}) \right\}_{i=1}^{L_f} \equiv \left\{ \left( \text{vec}(H_{i,f}^N) / \|h_{L+\delta,f}\|, h_{L+\delta,f}, \|h_{L+\delta,f}\| \right) \right\}_{i=1}^{L_f + N - 1}.
$$

(58)
As summarized in Table 3, we consider a window size $N = 5$ with lag size $\delta = 3$. All of the experimental results follow the 3GPP 5G standard SCM channel model [46] with variations of the long-term features over frames following Clause 7.6.3.2 (Procedure B) [46], under the Umi–Street Canyon environment, as discussed in Section 2.2. The normalized Doppler frequency $\rho = \gamma_{d,f} / \gamma_{SRS} \in [0, 1]$ within each frame $f$, defined as the ratio between the Doppler frequency $\gamma_{d,f}$ (4) and the frequency of the pilot symbols $\gamma_{SRS}$ or sounding reference signal (SRS) [46], is randomly selected in one of the two following ways: (i) for slow-varying environments, it is uniformly drawn in the interval $[0.005, 0.05]$; and (ii) for fast-varying environments, it is uniformly distributed in the interval $[0.1, 1]$. In the following, we study the impact of (i) the number of antennas $N_R N_T$, (ii) the number of channel taps $W$, (iii) the number of training samples $L_{\text{new}}$, and (iv) the number of previous frames $F$, for various prediction schemes: (a) conventional learning, (b) transfer learning, and (c) meta-learning, where each scheme is implemented by using either the naïve or the LSTD parametrization.

Table 3. Experimental setting.

| Window size ($N$) | 5 |
|-------------------|---|
| Lag size ($\delta$) | 3 |
| Number of previous frames ($F$) | 500 |
| Number of slots $(L + N - \delta + 1)$ | 107 |
| Frequency of the pilot signals $(\omega_{SRS}/2\pi)$ | 200 |
| Normalized Doppler frequency for slow-varying environment $\rho \sim \text{Unif}[0.005, 0.05]$ |
| Normalized Doppler frequency for fast-varying environment $\rho \sim \text{Unif}[0.1, 1]$ |
| SNR for channel estimation | 20 dB |
| Number of pilots for channel estimation | 100 |

5.1. Multi-Antenna Frequency-Flat Channels

We begin by considering multi-antenna frequency-flat channels and evaluating the NMSE as a function of total number of antennas $N_R N_T$ under a fast-varying environment (Figure 4) or a slow-varying environment (Figure 5). We set $K = 1$ in the LSTD model. Specific antenna configurations are described in Appendix E. Both transfer and meta-learning are seen to provide significant advantages as compared to conventional learning, as long as one chooses the type of parametrization—naïve or LSTD—as a function of the type of variability in the channel, with meta-learning generally outperforming transfer learning. In particular, as seen in Figure 4, for fast-varying environments, meta-learning with LSTD parametrization has the best performance, significantly reducing the NMSE with respect to both conventional and transfer learning. This is because meta-learning with LSTD can account for the need to adapt to fast-varying channel conditions, while also leveraging the reduced-rank structure of the channel. In contrast, as shown in Figure 5, for slow-varying channels, naïve parametrization tends to be preferable, because, as explained in Section 4.2, long-term and short-term features of the channel become indistinguishable when channel variability is too low. It is also interesting to observe that increasing the number of antennas is generally useful for prediction, as the predictor can build on a larger vector of correlated covariates. This is, however, not the case for conventional learning in slow-varying environments, for which the features tend to be too correlated, resulting in overfitting. As a final note, although absolute NMSE values close to 1 may be insufficient for use in applications such as precoding, they can provide useful information for other applications such as proactive resource allocation [40,64].
5.2. Rank Estimation

In the previous experiments, we have considered channels with unitary rank, for which one can assume without loss of optimality a number of features in the LSTD parametrization equal to $K = 1$. In order to implement predictors for multi-antenna frequency-selective channels, one instead needs to first address the problem of estimating the number of features. Here, we evaluate the performance of the approach proposed in Section 4.7 for rank estimation. To this end, we set the number of antennas as $N_R = 8$ and $N_T = 8$, and consider the 19-clustered channel model with delay spread ratio 2. Figure 6 shows the NMSE evaluated on the meta-training, meta-validation, and meta-test data sets as a function of total number of features $K$. The meta-training set contains 20 frames, the meta-test 200 frames, and the meta-validation set 20 frames. The meta-training loss is monotonically decreasing with $K$, because a richer parametrization enables a closer fit of the training data. In contrast, both meta-test and meta-validation loss are optimized for an intermediate value of $K$. The main point of the figure is that the meta-validation loss, while only containing 20 frames, provides useful information to choose a value of $K$ that approximately minimizes the meta-test loss. In contrast, although we can see that $K = 3$ is a proper estimate of the channel rank for the considered set-up, AIC-based rank estimation gives the highly overestimated value $K = 200$, which deteriorates the prediction performance, as can be seen in Figure 6. Throughout the following experiments, we will follow the proposed procedure to select $K$ for meta-learning, whereas for all the other schemes, we adopt AIC-based rank estimation to determine $K$. 
Accordingly, as discussed in the previous subsection, meta-learning outperforms both power, following ITU-R report [65]. Figure 7 shows that the dependence on the delay spread of the channel is qualitatively similar to the dependence on the number of antennas in Figures 4 and 5, with the top of Figure 7 representing the performance under a fast-varying environment and the bottom figure depicting the NMSE for a slow-varying environment. Figures 4 and 5, with the top of Figure 7 representing the performance under a fast-varying environment and the bottom figure depicting the NMSE for a slow-varying environment.

Before considering multi-antenna frequency-selective channels, we first consider the impact of the level of frequency selectivity on the prediction of single-antenna frequency-selective channels. To this end, starting from 45 ns, we increase the delay spread by a multiplicative factor, and correspondingly also increase the number of taps by the same amount, which is referred to as delay spread ratio in Figure 6. The number of taps W is obtained as the smallest number of taps that contains more than 90% of the average channel power, following ITU-R report [65]. Figure 7 shows that the dependence on the delay spread of the channel is qualitatively similar to the dependence on the number of antennas in Figures 4 and 5, with the top of Figure 7 representing the performance under a fast-varying environment and the bottom figure depicting the NMSE for a slow-varying environment. Accordingly, as discussed in the previous subsection, meta-learning outperforms both transfer and conventional learning, as long as the parametrization is correctly selected: naïve for slow-varying channels, and LSTD for fast-varying environments.

Figure 6. Multi-antenna frequency-selective channel prediction performance as a function of the number of features K, under 19-clustered, multi-taps (W = 4), multi-antenna (NT = 8, NR = 8) 3GPP SCM channel model for a fast-varying environment with number of training samples Lnew = 1. Results are evaluated with number of previous frames Ftr = 20 for meta-training, Fval = 20 for meta-validation, and Fte = 200 for meta-test.

5.3. Single-Antenna Frequency-Selective Channels

Figure 7. Single-antenna frequency-selective channel prediction performance as a function of delay spread ratio, under 19-clustered, multi-taps, single-antenna (NT = 1, NR = 1) 3GPP SCM channel model for a fast-varying environment (top) and slow-varying environment (bottom) with number of training samples Lnew = 1 (K = 1).
5.4. Multi-Antenna Frequency-Selective Channel Case

We now consider the prediction performance for multi-antenna frequency-selective channels as a function of the number of training samples $L_{\text{new}}$ in Figures 8 and 9, as well as versus the number of frames $F$ in Figure 10. For meta-learning, we set $L_{\text{tr}} = L_{\text{new}}$ in order to avoid discrepancies between meta-training and meta-testing [29]. Figure 8 and Figure 9 shows that meta-learning and transfer learning, which utilize $F = 500$ previous frames, can significantly outperform conventional learning in terms of number of required pilots $L_{\text{new}}$. This key observation motivates the use of transfer and meta-learning in the presence of limited training data. Furthermore, confirming the analysis in Section 3.3 and Section 4.6, meta-learning can outperform all other schemes as long as one selects a naive parametrization for slowly varying environments, and the LSTD parametrization for fast-varying environments. For sufficiently large $L_{\text{new}}$, transfer learning can, however, improve over meta-learning on fast-varying environments, as seen in Figure 8. This stems from the split of training and testing set applied by meta-learning, which can lead to a performance loss as $L_{\text{new}}$ increases.

![Figure 8](image1.png)

**Figure 8.** Multi-antenna frequency-selective channel prediction performance as a function of the number of training samples $L_{\text{new}}$, under 19-clustered, two taps ($W = 2$), multi-antenna ($N_T = 4$, $N_R = 2$) 3GPP SCM channel model for a fast-varying environment with total number of features $K = 2$ unless determined by Section 5.2.

![Figure 9](image2.png)

**Figure 9.** Multi-antenna frequency-selective channel prediction performance as a function of the number of training samples $L_{\text{new}}$, under 19-clustered, two taps ($W = 2$), multi-antenna ($N_T = 4$, $N_R = 2$) 3GPP SCM channel model for a slow-varying environment.
Lastly, we investigate the effect of the number of previous frames \(F\) for transfer and meta-learning. As a general result, as demonstrated by Figure 10, an increase in the number \(F\) of previous frames results in better performance for both transfer and meta-learning. Furthermore, in a slow-varying environment with a small value of \(F\), transfer learning can outperform meta-learning due to the limited need for adaptation, whereas meta-learning with the correctly select type of parametrization, outperforms transfer learning otherwise.

6. Conclusions

In this paper, we have introduced data-driven channel prediction strategies for multi-antenna frequency-selective channels that aim at reducing the number of pilots by integrating transfer and meta-learning with a novel parametrization of linear predictors. The methods leverage the underlying structure of the wireless channels, which can be expressed in terms of a long short-term decomposition (LSTD) into long-term space-time features and fading amplitudes. To enable transfer and meta-learning under an LSTD-based model, we have proposed an optimization strategy based on equilibrium propagation (EP) and alternating least squares (ALS). Numerical experiments have shown that the proposed LSTD-based transfer and meta-learning methods far outperform conventional prediction methods, especially in the few-pilots regime. For instance, under a standard 3GPP SCM channel model, assuming four transmit antennas and two receive antennas, using only one pilot meta-learning with LSTD can reduce the normalized prediction MSE by 3 dB as compared to standard learning techniques. Future work may consider the joint use of deep neural networks, in lieu of linear prediction filters, although related results for multi-antenna frequency-flat channels have not reported any significant advantage to date [14–16,19].

Author Contributions: Conceptualization, software, formal analysis, writing, S.P.; conceptualization, supervision, writing, project administration, funding acquisition, O.S. All authors have read and agreed to the published version of the manuscript.

Funding: The work of S.P and O.S. was partly supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 725731).
Appendix A. Derivation for Long-Short-Term-Decomposition (LSTD)-Based Predictor $V_f^{(K)}$

Because the $k$-th amplitudes $\hat{d}_{i-N+1,f}^k, \ldots, \hat{d}_{i+\delta,f}^k$ are all scalar values, we can write $\text{vec}([\hat{d}_{i,f}^k, \ldots, \hat{d}_{i-N+1,f}^k]) = [\hat{d}_{i,f}^k, \ldots, \hat{d}_{i-N+1,f}^k]^\top$ and $\hat{d}_{i+\delta,f}^k = (\hat{d}_{i+\delta,f}^k)^\top$. By using these equalities, we can plug in (30) and (31) to (32), to get the expression of the predicted Predictor $V_f$ from which we can easily obtain LSTD-based predictor matrix $V_f^{(K)} = \sum_{k=1}^K v_f^k \otimes (\hat{b}_f^k (\hat{b}_f^k)^\dagger)$.

Appendix B. Details on Conventional Learning for LSTD-Based Prediction

Recalling from (3) that $X_f^\dagger = [\text{vec}(H_1^N)^\dagger, \ldots, \text{vec}(H_{L_{tr},f}^N)^\dagger]^\top$, we can rewrite one part of ALS (42) in the form of standard ridge regression formula as

$$v_f^k \leftarrow \arg\min_{v_f^k} \left\{ \left\| X_f^\dagger (V_f^{(K)})^k - (Y_f^\dagger)^k \right\|_F^2 + \lambda_2 \left\| v_f^k - \bar{v}_f^k \right\|_2^2 \right\}$$

$$(v_f^{(K)})^k = v_f^k \otimes (\hat{b}_f^k (\hat{b}_f^k)^\dagger) = \arg\min_{v_f^k} \left\{ \sum_{i=1}^{L_{tr}} \left\| b_f^i (\hat{b}_f^i)^\dagger H_{1,i,f}^N ((v_f^{(K)})^\dagger)^\top - (y_{i,f}^N)^k \right\|_2^2 + \lambda_2 \left\| v_f^k - \bar{v}_f^k \right\|_2^2 \right\}$$

(A2)

with $(y_{i,f}^N)^k$ being the Hermitian transposition of the $i$-th row of the $k$-th residual target matrix $(Y_f^N)^k$ defined in (39), which can be solved in closed-form similar to (14) as

$$((v_f^{(K)})^\dagger)^\top = \left( \sum_{i=1}^{L_{tr}} (H_{1,i,f}^N)^\dagger b_f^i (\hat{b}_f^i)^\dagger H_{1,i,f}^N + \lambda_2 I \right)^{-1} \left( \sum_{i=1}^{L_{tr}} (H_{1,i,f}^N)^\dagger b_f^i (\hat{b}_f^i)^\dagger (y_{i,f}^N)^k + \lambda_2 \bar{v}_f^k \right).$$

(A3)

The initialization of the parameter $\hat{b}_f^k$ is set to the available hyperparameter $\bar{b}_f^k$.

Similarly, the other part of ALS (43) can be rewritten as

$$\hat{b}_f^k \leftarrow \arg\min_{\hat{b}_f^k} \left\{ \sum_{i=1}^{L_{tr}} \left\| b_f^i (\hat{b}_f^i)^\dagger H_{1,i,f}^N ((v_f^{(K)})^\dagger)^\top - (y_{i,f}^N)^k \right\|_2^2 + \lambda_1 \text{tr} \left( (\hat{b}_f^k)^\dagger (\hat{b}_f^k)^\dagger \hat{b}_f^k \right) \right\}$$

subject to $b_f^k b_k = 1$,

$$= \arg\min_{\hat{b}_f^k} \left\{ ((\hat{X}_{k,f})^\dagger)^\top (\hat{X}_{k,f}) - (\hat{Y}_{k,f})^\dagger (\hat{X}_{k,f}) - (\hat{X}_{k,f})^\dagger (\hat{Y}_{k,f}) \hat{b}_f^k \right\}$$

subject to $b_f^k b_k = 1$,  

(A4)
for which the solution of (A4) can be obtained by taking the eigenvector of \((\bar{X}_{k,f})^T(\bar{X}_{k,f}) - (\bar{Y}_{k,f})^T(\bar{X}_{k,f}) - (\bar{X}_{k,f})^T(\bar{Y}_{k,f}))\) that corresponds to the smallest eigenvalue, with the matrices \(\bar{X}_{k,f}\) and \(\bar{Y}_{k,f}\) defined as

\[
\bar{X}_{k,f} = \left(\frac{\bar{X}_{Y}^T k}{\sqrt{\lambda_1(\bar{b}^k)^T}}\right), \quad \bar{Y}_{k,f} = \left(\frac{\bar{Y}_{Y}^T k}{\sqrt{\lambda_1(\bar{b}^k)^T}}\right),
\]

where we denote \((\bar{X}_{Y}^T)^k = [(\bar{x}_{1,f}^k)^T, \ldots, (\bar{x}_{N,F,f}^k)^T]^T\) given \(\bar{x}_{i,f}^k = H_{i,F}((\bar{v}_f^k)^T)^T\). Note that we arbitrarily started ALS with vector \(v^k_f\), as this ordering did not show any meaningful impact on the final results, as also reported in [66].

**Appendix C. Details on Transfer Learning for LSTD-Based Prediction**

Solution of (45) can be directly obtained with the tools in Appendix B given \(\lambda_1, \lambda_2 = 0\) and by substituting \(X_f^Y\) and \(Y_f^Y\) with \([X_1, \ldots, X_F]^T\) and \([Y_1, \ldots, Y_F]^T\), respectively.

**Appendix D. Details on Meta-Learning for LSTD-Based Prediction**

Before deriving (52) and (53), for ease of representation, let us define inner loss function \(L_{f}^{\text{inner}}\), outer loss function \(L_{f}^{\text{outer}}\), and total loss function \(L_{f}^{\text{total}}\) as

\[
L_{f}^{\text{inner}}(\hat{b}_f^k, v_f^k | b^k, \sigma^k) = \left\| \frac{X_f^Y (V_f^{(K)})^k}{(b_f^k)^T + b_f^k} - (Y_f^Y)^k \right\|_F^2 - \lambda_1 \left(\frac{(\hat{b}_f^k)^T (\hat{b}_f^k)^T}{(b_f^k)^T + b_f^k}\right) + \lambda_2 \|v_f^k - \sigma^k\|^2,
\]

\[
L_{f}^{\text{outer}}(\hat{b}_f^k, v_f^k) = \left\| \frac{X_f^Y (V_f^{(K)})^k}{(b_f^k)^T + b_f^k} - (Y_f^Y)^k \right\|_F^2,
\]

and

\[
L_{f}^{\text{total}}(\hat{b}_f^k, v_f^k | b^k, \sigma^k, \alpha) = L_{f}^{\text{inner}}(\hat{b}_f^k, v_f^k | b^k, \sigma^k) + \alpha L_{f}^{\text{outer}}(\hat{b}_f^k, v_f^k),
\]

respectively. Because (A6) is scale-invariant to \((\hat{b}_f^k)^T\hat{b}_f^k\) (recall that \((V_f^{(K)})^k = v_f^k \otimes ((\hat{b}_f^k)^T)^k\)), this can be considered as an unconstrained version of (38), i.e., \((\hat{b}_f, v_f^*)\) in (38) minimizes (A6). Analogously, (A8) can be considered as an unconstrained expression of (54) as \((\hat{b}_f, v_f^*)\) in (54) minimizes (A8).

Assuming that the conditions of implicit function theorem [67] are satisfied with respect to \((\hat{b}_f^k, v_f^k)\), from the finite differentiation method [60], we can write the gradients for meta-learning as

\[
\nabla_{\hat{b}_f} \sum_{f=1}^{F} \left\| X_f^Y (V_f^{(K)})^k - (Y_f^Y)^k \right\|_F^2 = \lim_{\alpha \to 0} \sum_{f=1}^{F} \frac{1}{\alpha} \left( \frac{\partial L_{f}^{\text{total}}}{\partial \hat{b}_f} (\hat{b}_f^k, v_f^k | b^k, \sigma^k, \alpha) - \frac{\partial L_{f}^{\text{total}}}{\partial b^k} (\hat{b}_f^k, v_f^k | b^k, \sigma^k, 0) \right) = \lim_{\alpha \to 0} \frac{2}{\alpha} \sum_{f=1}^{F} \left( \hat{b}_f^k - \hat{b}_f(\hat{b}_f^k)^T \right),
\]

(A9)
and

\[ \nabla_a \sum_{f=1}^{F} \left\| X_{f}^{\text{te}} (V_{f}^{(k)})_{k,a} - (Y_{f}^{\text{te}})_{k,a} \right\|^2 \]

\[ = \lim_{a \to 0} \sum_{f=1}^{F} \frac{1}{\alpha} \left( \frac{\partial C^{\text{total}}}{\partial \alpha} (p_{f,a}, v_{f,a} | b_{f}, \rho_{f,a}, \alpha) - \frac{\partial C^{\text{total}}}{\partial \alpha} (p_{f,a}^{*}, v_{f,a}^{*} | b_{f}, \rho_{f,a}, 0) \right) \]

\[ = \lim_{a \to 0} \frac{2 \lambda_2}{\alpha} \sum_{f=1}^{F} (v_{f,a}^{*} - v_{f,a}^{*}) \]

which concludes the derivation of (52) and (53). A generalized proof along with useful characteristics of EP can be found in [60]. For initialization, the hyperparameter \( \hat{b} \) is set as the one-hot vector at position \( k \); while \( \hat{\sigma} \) are chosen as all-zero vectors [52].

### Appendix E. Details on the Antenna Configuration in Section 5.1

Following the table contains the specification of the antenna configurations in Section 5.1. We denote \((N_{R}^{\text{hor}}, N_{R}^{\text{ver}}, N_{R}^{\text{pol}}, N_{T}^{\text{hor}}, N_{T}^{\text{ver}}, N_{T}^{\text{pol}})\) by the pair of number of horizontal receive antennas \(N_{R}^{\text{hor}}\), number of vertical receive antennas \(N_{R}^{\text{ver}}\), number of polarizations of receive antennas \(N_{R}^{\text{pol}}\), number of horizontal transmit antennas \(N_{T}^{\text{hor}}\), number of vertical transmit antennas \(N_{T}^{\text{ver}}\), and number of polarizations of transmit antennas \(N_{T}^{\text{pol}}\). Note that \(N_{R}N_{T} = N_{R}^{\text{hor}}N_{R}^{\text{ver}}N_{R}^{\text{pol}}, N_{T}^{\text{hor}}N_{T}^{\text{ver}}N_{T}^{\text{pol}}\).

### Table A1. Antenna Configurations for Section 5.1.

| Number of Total Antennas \((N_{R}N_{T})\) | Antenna Configuration \((N_{R}^{\text{hor}}, N_{R}^{\text{ver}}, N_{R}^{\text{pol}}, N_{T}^{\text{hor}}, N_{T}^{\text{ver}}, N_{T}^{\text{pol}})\) |
|----------------------------------------|--------------------------------------------------|
| 1                                     | \((1,1,1,1,1,1)\)                              |
| 2                                     | \((1,1,1,2,1,1)\)                              |
| 4                                     | \((1,1,2,2,1,1)\)                              |
| 8                                     | \((2,1,1,2,2,1)\)                              |
| 16                                    | \((2,1,1,4,2,2)\)                              |
| 32                                    | \((2,1,1,4,2,2)\)                              |
| 64                                    | \((2,1,1,4,2,2)\)                              |
| 128                                   | \((2,2,1,4,2,2)\)                              |

### References

1. Tang, F.; Kawamoto, Y.; Kato, N.; Liu, J. Future intelligent and secure vehicular network toward 6G: Machine-learning approaches. *Proc. IEEE 2019*, *108*, 292–307. [CrossRef]
2. Hoeber, P.; Kaiser, S.; Robertson, P. Two-dimensional pilot-symbol-aided channel estimation by Wiener filtering. In Proceedings of the 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Munich, Germany, 21–24 April 1997.
3. Baddour, K.E.; Beaulieu, N.C. Autoregressive modeling for fading channel simulation. *IEEE Trans. Wirel. Commun.* 2005, *4*, 1650–1662. [CrossRef]
4. Duel-Hallen, A.; Hu, S.; Hallen, H. Long-range prediction of fading signals. *IEEE Signal Process. Mag.* 2000, *17*, 62–75. [CrossRef]
5. Liu, L.; Feng, H.; Yang, T.; Hu, B. MIMO-OFDM wireless channel prediction by exploiting spatial-temporal correlation. *IEEE Trans. Wirel. Commun.* 2013, *13*, 310–319. [CrossRef]
6. Min, C.; Chang, N.; Cha, J.; Kang, J. MIMO-OFDM downlink channel prediction for IEEE802.16e systems using Kalman filter. In Proceedings of the 2007 IEEE Wireless Communications and Networking Conference (WCNC), Hong Kong, China, 11–15 March 2007; pp. 942–946.
7. Komnakis, C.; Fragouli, C.; Sayed, A.H.; Wesel, R.D. Multi-input multi-output fading channel tracking and equalization using Kalman estimation. *IEEE Trans. Signal Process.* 2002, *50*, 1065–1076. [CrossRef]
8. Kashyap, S.; Mollén, C.; Björnsson, E.; Larsson, E.G. Performance analysis of (TDD) massive MIMO with Kalman channel prediction. In Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, 5–9 March 2017; pp. 3554–3558.
9. Simmons, N.; Gomes, S.B.F.; Yacoub, M.D.; Simeone, O.; Cotton, S.L.; Simmons, D.E. AI-Based Channel Prediction in D2D Links: An Empirical Validation. *IEEE Access* 2022, *10*, 65459–65472. [CrossRef]
10. Simon, D.; Shmaliy, Y.S. Unified forms for Kalman and finite impulse response filtering and smoothing. *Automatica* 2013, 49, 1892–1899. [CrossRef]

11. Shmaliy, Y.S. Linear optimal FIR estimation of discrete time-invariant state-space models. *IEEE Trans. Signal Process.* 2010, 58, 3086–3096. [CrossRef]

12. Pratik, K.; Amjad, R.A.; Behboodi, A.; Soriaga, J.B.; Welling, M. Neural Augmentation of Kalman Filter with Hypernetwork for Channel Tracking. *arXiv* 2021, arXiv:2109.12561.

13. Liu, W.; Yang, L.L.; Hanzo, L. Recurrent neural network based narrowband channel prediction. In *Proceedings of the 2006 IEEE 63rd Vehicular Technology Conference*, Melbourne, VIC, Australia, 7–10 May 2006; Volume 5, pp. 2173–2177.

14. Jiang, W.; Schotten, H.D. A comparison of wireless channel predictors: Artificial Intelligence versus Kalman filter. In *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 20–24 May 2019; pp. 1–6.

15. Jiang, W.; Strufe, M.; Schotten, H.D. Long-range MIMO channel prediction using recurrent neural networks. In *Proceedings of the 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, Las Vegas, NV, USA, 10–13 January 2020.

16. Kibugi, J.; Ribeiro, L.N.; Haardt, M. Machine Learning-Based Prediction of Time-Varying Rayleigh Channels. *arXiv* 2021, arXiv:2103.06131.

17. Yuan, J.; Ngo, H.Q.; Matthaiou, M. Machine learning-based channel prediction in massive MIMO with channel aging. *IEEE Trans. Wirel. Commun.* 2020, 19, 2960–2973. [CrossRef]

18. Zhang, Y.; Alkhateeb, A.; Madadi, P.; Jeon, J.; Cho, J.; Zhang, C. Predicting Future CSI Feedback For Highly-Mobile Massive MIMO Systems. *arXiv* 2022, arXiv:2202.02492.

19. Kim, H.; Kim, S.; Lee, H.; Jang, C.; Choi, Y.; Choi, J. Massive MIMO channel prediction: Kalman filtering vs. machine learning. *IEEE Trans. Commun.* 2020, 69, 518–528. [CrossRef]

20. Bogale, T.E.; Wang, X.; Le, L.B. Adaptive channel prediction, beamforming and scheduling design for 5G V2I network: Analytical and machine learning approaches. *IEEE Trans. Veh. Technol.* 2020, 69, 5055–5067. [CrossRef]

21. Torrey, L.; Shavlik, J. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*; IGI Global: Hershey, PA, USA, 2010.

22. Thrun, S.; Pratt, L. *Learning to Learn*; Springer Science & Business Media: Berlin, Germany, 2012.

23. Raghu, A.; Raghu, M.; Bengio, S.; Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv* 2019, arXiv:1909.01517.

24. Jose, S.T.; Park, S.; Simeone, O. Information-Theoretic Analysis of Epistemic Uncertainty in Bayesian Meta-learning. In *Proceedings of the AISTATS*, Virtual Event, 28–30 March 2022.

25. Yuan, Y.; Zheng, G.; Wong, K.K.; Ottersten, B.; Luo, Z.Q. Transfer learning and meta learning-based fast downlink beamforming adaptation. *IEEE Trans. Veh. Technol.* 2020, 20, 1742–1755. [CrossRef]

26. Ge, Y.; Fan, J. Beamforming optimization for intelligent reflecting surface assisted MISO: A deep transfer learning approach. *IEEE Trans. Veh. Technol.* 2021, 70, 3902–3907. [CrossRef]

27. Yang, Y.; Gao, F.; Zhong, Z.; Ai, B.; Alkhateeb, A. Deep transfer learning-based downlink channel prediction for FDD massive MIMO systems. *IEEE Trans. Commun.* 2020, 68, 7485–7497. [CrossRef]

28. Parera, C.; Redondi, A.E.; Cesana, M.; Liao, Q.; Malanchini, I. Transfer learning for channel quality prediction. In *Proceedings of the 2019 IEEE International Symposium on Measurements & Networking (M&N)*, Catania, Italy, 8–10 July 2019; pp. 1–6.

29. Park, S.; Jang, H.; Simeone, O.; Kang, J. Learning to demodulate from few pilots via offline and online meta-learning. *IEEE Trans. Signal Process.* 2020, 69, 226–239. [CrossRef]

30. Cohen, K.M.; Park, S.; Simeone, O.; Shamai, S. Learning to Learn to Demodulate with Uncertainty Quantification via Bayesian Meta-Learning. *arXiv* 2021, arXiv:2108.00785.

31. Mao, H.; Lu, H.; Lu, Y.; Zhu, D. RoemNet: Robust Meta Learning Based Channel Estimation in OFDM Systems. In *Proceedings of the ICC 2019—2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 20–24 May 2019.

32. Raviv, T.; Park, S.; Simeone, O.; Eldar, Y.C.; Shlezinger, N. Online Meta-Learning For Hybrid Model-Based Deep Receivers. *arXiv* 2022, arXiv:2203.14359.

33. Jiang, Y.; Kim, H.; Asnani, H.; Kannan, S. MIND: Model Independent Neural Decoder. In *Proceedings of the 2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Cannes, France, 2–5 July 2019.

34. Park, S.; Simeone, O.; Kang, J. Meta-learning to communicate: Fast end-to-end training for fading channels. In *Proceedings of the ICASSP 2020—2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 4–8 May 2020.

35. Park, S.; Simeone, O.; Kang, J. End-to-end fast training of communication links without a channel model via online meta-learning. In *Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Atlanta, GA, USA, 26–29 May 2020.

36. Goutay, M.; Aoudia, F.A.; Hoydis, J. Deep hypernetwork-based MIMO detection. In *Proceedings of the 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Atlanta, GA, USA, 26–29 May 2020.

37. Zhang, J.; Yuan, Y.; Zheng, G.; Krikidis, I.; Wong, K.K. Embedding Model Based Fast Meta Learning for Downlink Beamforming Adaptation. *IEEE Trans. Wirel. Commun.* 2021, 21, 149–162. [CrossRef]
38. Karasik, R.; Simeone, O.; Jang, H.; Shamai, S. Learning to Broadcast for Ultra-Reliable Communication with Differential Quality of Service via the Conditional Value at Risk. *Entropy* 2022, 24, 1363.

39. Hu, Y.; Chen, M.; Saad, W.; Poor, H.V.; Cui, S. Distributed multi-agent meta learning for trajectory design in wireless drone networks. *IEEE J. Sel. Areas Commun.* 2021, 39, 3177–3192. [CrossRef]

40. Nikoloska, I.; Simeone, O. Black-Box and Modular Meta-Learning for Power Control via Random Edge Graph Neural Networks. *arXiv* 2021, arXiv:2108.13178.

41. Simeone, O.; Spagnolini, U. Lower bound on training-based channel estimation error for frequency-selective block-fading Rayleigh MIMO channels. *IEEE Trans. Signal Process.* 2004, 52, 3265–3277. [CrossRef]

42. Cicerone, M.; Simeone, O.; Spagnolini, U. Channel estimation for MIMO-OFDM systems by modal analysis/filtering. *IEEE Trans. Commun.* 2006, 54, 2062–2074. [CrossRef]

43. Pedersen, K.I.; Andersen, J.B.; Kermoal, J.P.; Mogensen, P. A stochastic multiple-input-multiple-output radio channel model for evaluation of space-time coding algorithms. In Proceedings of the Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No. 00CH37152), Boston, MA, USA, 24–28 September 2000; Volume 2, pp. 893–897.

44. Abdi, A.; Kaveh, M. A space-time correlation model for multielement antenna systems in mobile fading channels. *IEEE J. Sel. Areas Commun.* 2002, 20, 550–560. [CrossRef]

45. Park, S.; Simeone, O. Predicting Flat-Fading Channels via Meta-Learned Closed-Form Linear Filters and Equilibrium Propagation. *arXiv* 2021, arXiv:2110.00414.

46. 3GPP. Study on Channel Model for Frequencies From 0.5 to 100 GHz (3GPP TR 38.901 Version 16.1.0 Release 16). TR 38.901. 2020. Available online: https://www.etsi.org/deliver/etsi_tr/138900_138999/138901/16.01.00_60/tr_138901v160100p.pdf (accessed on 16 September 2022).

47. Gallager, R.G. *Principles of Digital Communication*; Cambridge University Press: Cambridge, UK, 2008.

48. Simeone, O. A brief introduction to machine learning for engineers. *Found. Trends® Signal Process.* 2018, 12, 200–431. [CrossRef]

49. Simeone, O. *Machine Learning for Engineers*; Cambridge University Press: Cambridge, UK, 2022.

50. Prasad, R.; Murthy, C.R.; Rao, B.D. Joint approximately sparse channel estimation and data detection in OFDM systems using sparse Bayesian learning. *IEEE Trans. Signal Process.* 2014, 62, 3591–3603. [CrossRef]

51. Huang, C.; Liu, L.; Yuen, C.; Sun, S. Iterative channel estimation using LSE and sparse message passing for mmWave MIMO systems. *IEEE Trans. Signal Process.* 2018, 67, 245–259. [CrossRef]

52. Denevi, G.; Ciliberto, C.; Stamos, D.; Pontil, M. Learning to learn around a common mean. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, Montreal, QC, Canada, 3–8 December 2018.

53. Yin, M.; Tucker, G.; Zhou, M.; Levine, S.; Finn, C. Meta-learning without memorization. *arXiv* 2019, arXiv:1912.03820.

54. Swindlehurst, A.L. Time delay and spatial signature estimation using known asynchronous signals. *IEEE Trans. Signal Process.* 1998, 46, 449–462. [CrossRef]

55. Nicoli, M.; Simeone, O.; Spagnolini, U. Multislot estimation of fast-varying space-time communication channels. *IEEE Trans. Signal Process.* 2003, 51, 1184–1195. [CrossRef]

56. Cortes, C.; Mohri, M.; Rostamizadeh, A. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.* 2012, 13, 795–828.

57. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* 1987, 2, 37–52. [CrossRef]

58. Wong, A.S.Y.; Wong, K.W.; Leung, C.S. Distributed multi-agent learning for trajectory design in wireless drone networks. *IEEE J. Sel. Areas Commun.* 2021, 39, 3177–3192. [CrossRef]

59. Sidiropoulos, N.D.; De Lathauwer, L.; Fu, X.; Huang, K.; Papalexakis, E.E.; Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Trans. Signal Process.* 2017, 65, 3551–3582. [CrossRef]

60. Scellier, B.; Bengio, Y. Equilibrium propagation: Bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* 2017, 11, 24. [CrossRef]

61. Zucchet, N.; Schug, S.; von Oswald, J.; Zhao, D.; Sacramento, J. A contrastive rule for meta-learning. *arXiv* 2021, arXiv:2104.01677.

62. Wax, M.; Kailath, T. Detection of signals by information theoretic criteria. *IEEE Trans. Acoust. Speech Signal Process.* 1985, 33, 387–392. [CrossRef]

63. Liavas, A.P.; Regalia, P.A.; Delmas, J.P. Blind channel approximation: Effective channel order determination. *IEEE Trans. Signal Process.* 1999, 47, 3336–3344. [CrossRef]

64. Agrawal, A.; Andrews, J.G.; Cioffi, J.M.; Meng, T. Iterative power control for imperfect successive interference cancellation. *IEEE Trans. Wirel. Commun.* 2005, 4, 878–884. [CrossRef]

65. Recommendations, I. Multipath Propagation and Parameterization of its Characteristics. Available online: https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.1407-0-199907-S!!PDF-E.pdf (accessed on 22 September 2022).

66. Xiao, C.; Yang, C.; Li, M. Efficient alternating least squares algorithms for low multilinear rank approximation of tensors. *J. Sci. Comput.* 2021, 87, 1–25. [CrossRef]

67. Lorraine, J.; Vicol, P.; Duvenaud, D. Optimizing millions of hyperparameters by implicit differentiation. In Proceedings of the AISTATS, Virtual Event, 26–28 August 2020.