Low-Complexity Compressive Channel Estimation for IRS-Aided mmWave Systems With Hypernetwork-Assisted LAMP Network

Wen-Chiao Tsai, Chi-Wei Chen, Chieh-Fang Teng, Student Member, IEEE, and An-Yeu Wu, Fellow, IEEE

Abstract—Intelligent reflecting surface (IRS) can enhance the wireless communication environment by smartly reflecting the incident signal toward desired directions. However, the acquisition of channel state information (CSI) is challenging since IRS usually consists of a massive number of passive elements that have no capabilities of sensing and processing the pilot signals. Although by exploiting the sparsity of the angular domain channel, the huge pilot overhead can be reduced with conventional compressive sensing algorithms, such as approximate message passing (AMP), these algorithms cannot achieve satisfactory channel estimation performance. Based on the learned AMP (LAMP) network, we propose a hypernetwork-assisted LAMP (HN-LAMP) network with dynamic shrinkage parameters to improve the channel estimation accuracy. Furthermore, a recurrent architecture is adopted to reduce the large memory overhead arising from the LAMP network. Simulation results show that the proposed HN-LAMP network can improve the channel estimation accuracy or reduce the computational complexity under satisfactory estimation performance. Moreover, the proposed hypernetwork-assisted recurrent LAMP (HNR-LAMP) architecture can effectively reduce 50% memory overhead by sharing learnable weights.

Index Terms—Intelligent reflecting surface (IRS), channel estimation, compressive sensing, deep learning, hypernetwork.

I. INTRODUCTION

RECENTLY, intelligent reflecting surface (IRS) is emerging as a promising technology to significantly enhance the capacity, coverage, and energy efficiency of the wireless communication system [1]. In general, IRS consists of a large number of low-cost passive reflective elements, each of which can reflect the incident signal with a reconfigurable amplitude and phase shift. By smartly adjusting the phase shifts of the IRS elements, IRS can reflect the incident electromagnetic waves toward the desired directions, in effect providing beamforming gain between the transmitter and the receiver [2].

To realize the full potential of IRS, reliable channel state information (CSI) is required for the design of joint active and passive beamforming. However, the acquisition of CSI is challenging since the IRS elements are usually passive that have no capabilities of sensing and processing the pilot signals. In addition, since IRS usually consists of hundreds of elements, the large size of the cascaded channel requires high pilot overhead for channel estimation. Therefore, a channel estimation method with reduced pilot overhead and high estimation accuracy is required for the IRS-aided systems, which cannot be realized by most of the existing solutions. In [3], an optimal IRS activation pattern using array gain is derived for the least squares (LS) estimator, which achieves a lower estimation error compared with the ON/OFF-based method in [4]. However, the huge pilot overhead in [3] and [4] limits the potential performance improvement gained from the IRS. On the other hand, [5] and [6] can reduce the high pilot overhead by using compressive sensing (CS) to exploit the sparsity nature of the virtual angular domain (VAD) channel. Nevertheless, classical CS algorithms, such as orthogonal matching pursuit (OMP) [7] and approximate message passing (AMP) algorithms [8], cannot achieve satisfactory channel estimation accuracy since the sparsity of the angular cascaded channel is not significant [9].

By boosting the conventional model-based algorithm that has certain performance guarantees with the powerful deep learning (DL) tools, [10] proposed a learned AMP (LAMP) network from the deep unfolding of the AMP algorithm. By optimizing the linear transform matrices and the shrinkage parameter in each layer through DL, the LAMP network can significantly improve the estimation accuracy of the AMP algorithm even though the cascaded channel suffers from high sparsity level. However, two issues should be addressed:

1) Fixed shrinkage parameter for each layer: Although the optimized shrinkage parameters can be derived in a data-driven manner, the fixed shrinkage parameters cannot adapt to the current channel recovery status, which limits the potential of the LAMP network.

2) Large memory overhead: Due to the unfolding of the AMP algorithm and the large sizes of the linear transform matrices, the LAMP network suffers from large memory overhead, which hinders its deployment in practical systems.

In this letter, we propose a hypernetwork-assisted recurrent LAMP (HNR-LAMP) network to deal with the above issues. The main contributions are summarized as follows:

1) Dynamic shrinkage parameters via hypernetwork: Instead of using a fixed shrinkage parameter in each layer, we provide an adaptive shrinkage parameter through a hypernetwork, by considering the current channel recovery status, the estimation accuracy can be improved efficiently in different iterations and all the SNR region.

Manuscript received 30 March 2022; revised 28 April 2022; accepted 24 May 2022. Date of publication 26 May 2022; date of current version 12 August 2022. This work was supported by MedisTek, Inc., Hsinchu, Taiwan, under Grant MTKC-2022-0125. The associate editor coordinating the review of this letter and approving it for publication was C. Pan. (Corresponding author: An-Yeu Wu.)

The authors are with the Graduate Institute of Electronics Engineering and the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: daniel@access.ee.ntu.edu.tw; wilbur@access.ee.ntu.edu.tw; andywu@ntu.edu.tw). Digital Object Identifier 10.1109/LCOMM.2022.3178188
Fig. 1. An IRS-aided single-user wireless communication system.

2) Recurrent architecture for LAMP-based networks: By adopting a recurrent architecture, the weights in a layer are learned to share across iterations. Besides, an incremental weight sharing mechanism is proposed to determine the number of iterations for each layer. With the proposed HNR-LAMP, the memory overhead can be effectively reduced by 50% under satisfactory channel estimation accuracy.

Notations: The imaginary unit is denoted by $j \triangleq \sqrt{-1}$. $\mathbb{C}$ denotes the set of complex numbers. Lower and upper-case boldface letters denote column vectors and matrices, respectively. $\mathbf{A}^T$, $\mathbf{A}^H$, $\mathbf{A}^{-1}$, and $\text{vec} (\mathbf{A})$ are the transpose, conjugate transpose, inverse, and vectorization of matrix $\mathbf{A}$, respectively. $\|x\|$ denotes the $l_2$-norm of vector $x$, while $\text{diag}(x)$ denotes the diagonal matrix with $x$ as its main diagonal elements. The Kronecker product is represented by $\otimes$. Finally, $\mathcal{CN}(0, \Sigma)$ denotes the complex Gaussian distribution with zero mean and covariance matrix $\Sigma$, and $\mathbb{E}\{\cdot\}$ denotes the expectation operator.

II. SYSTEM MODEL

In this section, we will introduce the system model and the channel model of the IRS-aided mmWave communication systems. Then, the channel estimation problem will be formulated as a sparse signal recovery problem.

A. System Model and Channel Model

Consider an IRS-aided uplink wireless communication system as shown in Fig. 1, where a single-antenna user is served by a BS with $M$-element uniform linear array (ULA) and an IRS with $N$-element uniform planar array (UPA). The reflection coefficients of the IRS can be controlled by the BS through an IRS controller to reflect the incident signals toward desired directions.

Let $\mathbf{h}_d \in \mathbb{C}^{M \times 1}$ denote the direct channel from the user to the BS, $\mathbf{b}_r \in \mathbb{C}^{N \times 1}$ denote the channel from user to the IRS, and $\mathbf{G}^H \in \mathbb{C}^{M \times N}$ denote the channel from the IRS to the BS. The $t^{\text{th}}$ ($t = 1, \ldots, T$) time instant of a coherence interval, the received pilot signal at the BS can be expressed as

$$\mathbf{y}_t = (\mathbf{G}^H \text{diag}(\theta_d) \mathbf{h}_d + \mathbf{h}_r) x_t + \mathbf{n}_t,$$

where $x_t$ is the transmitted pilot signal, $\theta_d = [\theta_{d,1}, \ldots, \theta_{d,N}]^T$ is the reflecting vector at the IRS with $\theta_{d,n}$ representing the reflection coefficient associated with the $n^{\text{th}}$ element ($n = 1, \ldots, N$), and $\mathbf{n}_t \sim \mathcal{CN}(0, \sigma_n^2 \mathbf{I}_M)$ denotes the additive white Gaussian noise vector with the noise power $\sigma_n^2$. Note that the reflection coefficient can be written as $\beta_{d,n} = \beta_{d,n} e^{j\phi_{d,n}}$, with $\beta_{d,n} \in [0, 1]$ and $\phi_{d,n} \in [0, 2\pi)$ representing the controllable amplitude and phase shift, respectively. We assume $\beta_{d,n} = 1$ to avoid energy loss. Define $\mathbf{H}_c = \mathbf{G}^H \text{diag}(\mathbf{h}_r)$ as the cascaded channel, and we can rewrite (1) as

$$\mathbf{y}_t = (\mathbf{H}_c \mathbf{\Theta}_d + \mathbf{h}_r) x_t + \mathbf{n}_t.$$  

(2)

We adopt the widely used Saleh-Valenzuela channel model to characterize the channels $\mathbf{G}^H$ and $\mathbf{h}_r$ [9], and then $\mathbf{H}_c$ can be rewritten accordingly. Specifically, the channels $\mathbf{G}^H$, $\mathbf{h}_r$, and $\mathbf{H}_c$ can be modeled as

$$\mathbf{G}^H = \sqrt{\frac{M \times N}{L_G}} \sum_{l=1}^{L_G} \alpha_l \mathbf{a}_B (\varphi_l^G) \mathbf{a}_R^H (\varphi_l^G, \gamma_l^G),$$  

(3)

$$\mathbf{h}_r = \sqrt{\frac{N}{L_r}} \sum_{l=1}^{L_r} \alpha_P \mathbf{a}_R (\varphi_l^P, \gamma_l^P),$$  

(4)

$$\mathbf{H}_c = \sqrt{\frac{M \times N}{L_G L_r}} \sum_{l=1}^{L_G \times L_r} \alpha_l \alpha_P \mathbf{a}_B (\varphi_l^G) \mathbf{a}_R (\varphi_l^G, \gamma_l^G - \gamma_l^P).$$  

(5)

where $L_G$ is the number of IRS-BS paths, $L_r$ is the number of user-IRS paths, $\alpha_l(\alpha_P)$ denotes the complex path gain, $\varphi_l^G$ denotes the azimuth angle at the BS, $\varphi_l^G$ and $\varphi_l^P$ ($\gamma_l^G$ and $\gamma_l^P$) denote the azimuth (elevation) angles at the IRS, and $\mathbf{a}_B (\varphi)$ and $\mathbf{a}_R (\varphi)$ are the array response vectors at the IRS and the BS, respectively.

Due to the limited scattering nature of the mmWave channels, the number of cascaded paths $L_G L_r$ is much smaller than $M N$. Hence the cascaded channel exhibits sparsity in the VAD. By considering the VAD representation, we can approximate $\mathbf{H}_c$ as

$$\mathbf{H}_c \approx \mathbf{A}_B \widetilde{\mathbf{H}}_c \mathbf{A}_R^H,$$

(6)

where $\mathbf{H}_c \in \mathbb{C}^{M_G \times N_G}$ denotes the angular cascaded channel with resolutions $M_G$ and $N_G$, and $\mathbf{A}_B \in \mathbb{C}^{M \times M_G}$ ($\mathbf{A}_R \in \mathbb{C}^{N \times N_G}$) is the dictionary matrix consisting of the array response vectors $\mathbf{a}_B (\varphi)$ ($\mathbf{a}_R (\theta, \gamma)$) on a pre-discretized virtual angular grid. Note that we make $M_G \geq M$ and $N_G \geq N$ to well approximate the cascaded channel, and $\mathbf{H}_c$ is a sparse matrix.

B. Problem Formulation

In this letter, we assume that the direct channel is well approximated as in the conventional communication system without IRS, which can be realized by turning off the IRS elements. Hence, we will focus on the challenging cascaded channel estimation problem in the sequel. After $T$ instants of pilot transmission, we aggregate all the received signals at the BS by $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_T]$. After removing the effect of $\mathbf{h}_d$, $\mathbf{Y}$ can be expressed as

$$\mathbf{Y} = \mathbf{H}_c \Theta \mathbf{X} + \mathbf{N},$$

(7)

where $\Theta = [\Theta_1, \ldots, \Theta_T]$ is the reflection matrix for channel sensing, $\mathbf{X} = \text{diag}([x_1, \ldots, x_T])$, and $\mathbf{N} = [\mathbf{n}_1, \ldots, \mathbf{n}_T]$. By applying the vectorization operation on (7) to obtain the measurement vector $\mathbf{y} \in \mathbb{C}^{Q G \times 1}$ ($Q = MT$) and using $\mathbf{X} = \mathbf{I}_T$ as the transmitted pilots, we have

$$\mathbf{y} = \text{vec} (\mathbf{Y}) = (\Theta^T \otimes \mathbf{I}_M) \mathbf{h}_c + \mathbf{n} = (\Theta^T \otimes \mathbf{I}_M) (\mathbf{A}_R^* \otimes \mathbf{A}_B) \mathbf{h}_c + \mathbf{n} = \mathbf{W} \mathbf{h}_c + \mathbf{n},$$

(8)

where $\mathbf{h}_c = \text{vec}(\mathbf{H}_c)$, $\mathbf{h}_r = \text{vec}(\mathbf{H}_r)$, $\mathbf{n} = \text{vec}(\mathbf{N})$, and $\mathbf{W} = (\Theta^T \otimes \mathbf{I}_M) (\mathbf{A}_R^* \otimes \mathbf{A}_B)$ is the $Q \times M_G N_G$ measurement matrix.
Fig. 2. The architecture and the $i^{th}$ layer of the LAMP network.

Based on (8), we have formulated the cascaded channel estimation problem as a sparse signal recovery problem. However, since the number of cascaded paths $L_{C}L_{r}$ becomes larger due to the double scatters at the BS and the IRS, the sparsity of the angular cascaded channel is less significant compared with the angular channel in conventional communications. Moreover, the double-path-loss effect of the cascaded channel would also deteriorate the channel estimation performance. Consequently, conventional CS algorithms, such as OMP and AMP, cannot achieve satisfactory estimation accuracy with low pilot overhead. Even though the accuracy can be improved by adopting virtual angle grids with higher resolutions, the increase of computational complexity makes these methods infeasible in practical communication systems.

III. PROPOSED HYPERNETWORK-ASSISTED RECURRENT LAMP NETWORK FOR COMPRESSIVE CHANNEL ESTIMATION

In this section, we propose a hypernetwork-assisted LAMP (HN-LAMP) network to improve the channel estimation performance. Besides, a recurrent architecture is proposed to alleviate the large memory overhead of the LAMP network.

A. LAMP Network

Although AMP is a powerful algorithm for large-scale sparse signal recovery problems, the linear transform matrices are only selected for the general sparse signal recovery problem; and the shrinkage parameter takes the same empirical value for all iterations. By unfolding the iterations of the AMP algorithm into a LAMP network, the tunable parameters $\mathcal{W} = \{ W_i, Z_i, \alpha_i \}_{i=0}^{L-1}$ can be optimized based on the data distribution of the cascaded channels. Therefore, the limitations of the AMP algorithm can be relaxed, and the LAMP network can be more suitable for the problem of cascaded channel estimation. As illustrated in Fig. 2, the $i^{th}$ layer of the LAMP network estimates the angular cascaded channel as follows [10]

$$\hat{h}_{c,i+1} = \eta_{st}(\hat{h}_{c,i} + Z_i v_i; \frac{\alpha_i}{\sqrt{Q}} \| v_i \|), \quad (9)$$

$$v_{i+1} = y - W_i \hat{h}_{c,i+1} + \frac{1}{Q} \| \hat{h}_{c,i+1} \|_0 v_i, \quad (10)$$

where $\hat{h}_{c,i+1}$ and $v_{i+1}$ are the outputs of the $i^{th}$ layer and the inputs of the $(i+1)^{th}$ layer, representing the estimated angular channel and the residual noise vector, respectively. The soft thresholding denoiser $[\eta_{st}(x; \lambda)]_+ = \text{sgn}(x) \max \{ |x| - \lambda, 0 \}$ can make the estimated vector sparser by shrinking the amplitude of the low power elements to zero. Finally, we obtain the estimated cascaded channel $\hat{h}_{c}$ by transforming the angular channel into the spatial channel via $\hat{h}_{c} = (A_R^* \otimes A_D) \hat{h}_{c,L}$.

B. Hypernetwork Assisted Dynamic Shrinkage Parameters

Although the LAMP network can learn the optimized shrinkage parameters $\{ \alpha_{i,LAMP} \}_{i=0}^{L-1}$ based on the data distribution of the cascaded channels, the parameter is fixed for each layer and cannot adapt to the current channel recovery status. Therefore, we propose a HN-LAMP network to generate the shrinkage parameters with a small network, called hypernetwork. The hypernetwork allows the shrinkage parameter to be generated dynamically and adapt to the current channel recovery status by using the residual noise vector as shown in Fig. 3.

To share a single network among all the layers for low computational complexity, the dynamic shrinkage parameter $\alpha_{i,\text{hyper}}$ is divided into two parts. The static part $\alpha_{i}^{s}$ is optimized for the $i^{th}$ layer like the original LAMP network, where $\alpha_{i}^{s}$ is a general value for any cascaded channel and varies among layers. On the other hand, the dynamic part $\alpha_{i}^{d}$ is generated by the hypernetwork to only capture the variance in the cascaded channels, so that the shrinkage parameter $\alpha_{i,\text{hyper}}$ can be adaptive to a specific cascaded channel and the dynamic range of the hypernetwork can be reduced with the static part $\alpha_{i}^{s}$ of each layer. Specifically, $\alpha_{i,\text{hyper}}$ is obtained as follows

$$\alpha_{i,\text{hyper}} = \alpha_{i}^{s} + \alpha_{i}^{d} = \alpha_{i}^{s} + \hat{h}(\frac{v_i}{\| v_i \|}; \theta_{\text{hyper}}), \quad (11)$$

where the residual noise vector $v_i$ is the input of the hypernetwork to consider the current channel recovery status, and $\theta_{\text{hyper}}$ contains the parameters for the hypernetwork. Note that a clipped ReLU activation function $\sigma(x, \alpha_m)$ is used to limit the dynamic range of $\alpha_{i}^{d}$ in the region $[0, \alpha_m]$. The clipped ReLU function is very useful to guarantee stable convergence and can be expressed as

$$\sigma(x, \alpha_m) = \begin{cases} \alpha_m, & x > \alpha_m, \\ x, & 0 < x \leq \alpha_m, \\ 0, & \text{otherwise}. \end{cases} \quad (12)$$

The training process is divided into two steps. We first train the original LAMP network, and then the tunable parameters $\mathcal{W}$ are used to initialize the HN-LAMP network, where the shrinkage parameter is used to initialize the static component with $\alpha_{i}^{s} = \alpha_{i,LAMP}$. In the second step, all the parameters $\alpha_{i,\text{hyper}} = \{ W_i, Z_i, \alpha_{i}^{s}, v_{i}, \theta_{\text{hyper}} \}$ are trained together in an end-to-end manner, and the hypernetwork learns how to generate the dynamic components for all the HN-LAMP layers.

However, there are still two issues that need to be addressed regarding the LAMP network. That is the fixed shrinkage parameter $\alpha_{i,LAMP}$ for each LAMP layer and the large memory overhead of the LAMP network, which will be tackled in Section III-B and III-C, respectively.

Fig. 3. The hypernetwork architecture for the shrinkage parameters $\{ \alpha_{i,\text{hyper}} \}_{i=0}^{L-1}$ in the $i^{th}$ layer of the HN-LAMP network.
C. Recurrent Architecture With Incremental Weight Sharing Mechanism

Although the LAMP network greatly improves the estimation accuracy by relaxing the iterations of the AMP into trainable layers, the additional memory overhead arising from the tunable parameters $\mathcal{W}$ and $\mathcal{W}^{\text{hyper}}$ hinders the realizability of the networks for practical IRS-aided communication systems. Therefore, we propose the recurrent LAMP (R-LAMP) network to share the parameters of the LAMP network among different layers. Besides, we can form the HN-R-LAMP network by providing dynamic shrinkage parameters for the R-LAMP based on the hypernetwork architecture in Fig. 3.

The idea of weight sharing with a recurrent architecture has been used in decoding BCH codes [11] and polar codes [12], which can significantly reduce the number of parameters with negligible performance degradation. Unfortunately, the naïve recurrent LAMP with only a single layer of parameters $\{\mathbf{W}_0, \alpha_0^{\text{LAMP}}\}$ for all the iterations suffers from severe degradation of the estimation accuracy. In other words, the heterogeneity of the parameters for different layers is necessary to achieve a satisfactory estimation accuracy.

Fig. 4 shows the architecture of the R-LAMP network. To maintain the performance of cascaded channel estimation, we consider retaining the heterogeneity of the parameters with $L'$ recurrent blocks, where we make $L' < L$ to reduce the high memory overhead in the original LAMP network. To maximize the benefit of a recurrent block, an incremental weight sharing mechanism is proposed to determine the number of iterations $n_i$ of each recurrent block in the offline training stage.

Algorithm 1 shows the proposed incremental weight sharing mechanism. In the mechanism, different architectures of the R-LAMP network are evaluated based on their performance on the validation dataset $\mathcal{D}_{\text{valid}}$ of $D_v$ validation data. The validation loss is the normalized mean square error (NMSE) defined as $L_v = \frac{1}{D_v} \sum_{c=1}^{D_v} \left\{ \| \mathbf{h}_c, i - \hat{\mathbf{h}}_c, i \|^2 / \| \mathbf{h}_c, i \|^2 \right\}$. During the training procedure, the best loss $L_b$ is used to denote the performance of the current architecture, and $L_b$ will be updated if the attempt to increase an iteration improves the channel estimation performance with a lower validation loss. Step 1 learns the R-LAMP network, where a single iteration is used for each recurrent block. Steps 2-9 represent the sequential training sub-procedure from $i = 0$ to $L' - 1$. Steps 3-7 continuously attempt to increase the number of iterations $n_i$ for the $i$th recurrent block if $L_v$ improves in the current attempt. Otherwise, Step 8 decreases $n_i$ by one since the previous attempt fails to improve the channel estimation performance.

IV. SIMULATION RESULTS

In our simulations, the number of BS antennas is $M = 16$, and the number of IRS elements is $N = 64$. The numbers of paths are set to $L_G = L_r = 3$. The azimuth/elevation angles are uniformly generated from $(-\pi/2, \pi/2)$, which may not lie on the pre-discretized virtual angular grids. We consider practical path gains in IRS systems $\alpha_l \sim \mathcal{CN}(0,10^{-3}d_{BR}^{-2})$ and $\alpha_v \sim \mathcal{CN}(0,10^{-3}d_{BR}^{-2})$, where the distance between the BS and IRS and the distance between the IRS and user are assumed to be $d_{BR} = 10m$ and $d_{BR} = 100m$, respectively. The SNR is defined as $\mathbb{E}\{ \| \mathbf{H}, \Theta \|_F^2 / \| \mathbf{N} \|_F^2 \}$ in (7), where the elements of $\Theta$ follow the $\mathcal{U}(0,2\pi)$ distribution.

For the proposed LAMP-based networks, we generate $10^5$ channel samples for training. For the OMP algorithm, we consider various angle grid resolutions, indicated by the over-sampling rates $\beta = M_G/M = \sqrt{N_G/N}$. Besides, the double-structured orthogonal matching pursuit (DS-OMP) algorithm [9] is compared, where we only consider $\beta = 1$ since DS-OMP assumes unitary dictionary matrices. The NMSE metric is defined as $\text{NMSE} = \mathbb{E}\{ \| \mathbf{h}_c - \hat{\mathbf{h}}_c \|_2^2 / \| \mathbf{h}_c \|_2^2 \}$.

A. Comparison of Performance for Proposed Networks

Fig. 5(a) and Fig. 5(b) show the NMSE performance comparison against SNR under the time slots (pilot overhead) $T = 32$. The OMP algorithm with $\beta = 1$ suffers from the power leakage effect and cannot achieve satisfactory channel estimation accuracy. However, the effect of power leakage can be suppressed by adopting higher oversampling rates ($\beta = 2, 4$) under the price of unfeasible computational complexity. Besides, although the DS-OMP algorithm can achieve the lowest computational complexity through the decomposition method, it also suffers from the power leakage effect and has slightly worse NMSE than the OMP algorithm with $\beta = 1$. By relaxing the AMP algorithm with deep unfolding, the LAMP-based networks greatly improve the channel estimation performance of AMP algorithm and outperform the OMP algorithm with $\beta = 2$. Furthermore, the HN-LAMP networks outperform their counterpart LAMP networks in all SNRs and the OMP algorithm with $\beta = 4$ in the low SNR region, which shows the effectiveness of the hypernetwork to provide dynamic shrinkage parameters.

Fig. 5(b) shows the performance of the R-LAMP and HN-R-LAMP networks. The number of the recurrent blocks $L'$ is set to 5. The LAMP network with 5 layers is the initial
architecture for the incremental weight sharing mechanism. After completing the training procedure, the numbers of iterations for the recurrent blocks are $\{n_i\}_{i=0} = \{5, 1, 1, 1, 2\}$. The final R-LAMP network can reduce the memory overhead by 50% since it achieves the performance of the LAMP with 10 layers. Moreover, the HNR-LAMP network can further lower the estimation error and outperform the LAMP network in the low SNR region.

Fig. 5(c) shows the NMSE performance comparison versus pilot overhead, indicated by the number of time instants $T$. As shown in Fig. 5(c), in order to achieve the same estimation accuracy, we can observe that the proposed HN-LAMP network and HNR-LAMP require much lower pilot overhead compared with the OMP with $\beta = 1$ and the DS-OMP algorithm, which demonstrates the ability of the proposed networks to achieve high estimation accuracy under low pilot overhead.

B. Analysis of Computational Complexity

In this letter, we develop the HN-LAMP and HNR-LAMP networks to estimate the cascaded channel with low complexity. Specifically, the HN-LAMP can improve the channel estimation accuracy by providing dynamic shrinkage parameters. Besides, the HNR-LAMP network can further reduce the large memory overhead in the LAMP-based network. Simulation results show that the proposed HN-LAMP network with dynamic shrinkage parameters can achieve superior performance under lower computational complexity. Moreover, the HNR-LAMP network can effectively reduce the memory overhead by 50% and achieve satisfactory channel estimation performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Computational Complexity</th>
<th>Execution Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMP</td>
<td>$O(\beta^3 MNQ L_0L_r) + O(Q(L_0L_r)^3)$</td>
<td>24.1 ($\beta = 1$), 184.6 ($\beta = 2$), 1994.5 ($\beta = 4$)</td>
</tr>
<tr>
<td>DS-OMP</td>
<td>$O(MQ) + O(L_0NQZ^2)$</td>
<td>1.3</td>
</tr>
<tr>
<td>LAMP</td>
<td>$O(LQMN)$</td>
<td>11.4 ($L = 10$)</td>
</tr>
<tr>
<td>HN-LAMP</td>
<td>$O(LQMN) + O(LQN_h)$</td>
<td>12.2 ($L = 10$)</td>
</tr>
<tr>
<td>HNR-LAMP</td>
<td>$O(n_TQMN) + O(n_TQN_h)$</td>
<td>12.2 ($n_T = 10$)</td>
</tr>
</tbody>
</table>

V. Conclusion

B. Analysis of Computational Complexity

Table I summarizes the comparison of computational complexity. For practical implementation, the computation time on an AMD Ryzen Threadripper 2990WX CPU is also specified. For a HN-LAMP network with $L$ layers, the computational complexity is $(LQMN) + O(LQN_h)$ for the computation of the linear transforms and the hypernetwork with $N_h$ hidden nodes. Similarly, the complexity of the HN-RAMP network can be obtained by considering the total iterations $n_T$. As shown in Fig. 5(a) and Fig. 5(b), the proposed HN-LAMP and HNR-LAMP networks outperform OMP with $\beta = 2$ and OMP with $\beta = 4$ in the low SNR region, which can effectively reduce the execution time by 93% and 99%, respectively.

REFERENCES