Scenario-Aware Learning Approaches to Adaptive Channel Estimation

Runhua Li†, Jian Sun†, Member, IEEE, Jiang Xue†, Senior Member, IEEE, Christos Masouros‡, Senior Member, IEEE
† Xi’an Jiaotong University, Xi’an, China.
‡ University College London, London, U.K.

Abstract—The growth of frequency bandwidths and applications with the forthcoming generations of wireless networks will give rise to a multitude of wireless transmission scenarios, topologies and channel structures. In this work we go beyond existing learning-based channel estimation methods tailored for specific scenarios, to develop an adaptive learning-based channel state information (CSI) estimation approach. We offer the adaptivity in the learning approach through extracting the statistical information of CSI and adjusting the channel estimation method with the extracted information automatically in each scenario. Specifically, Learning-Based Scenario-Adaptive Channel Estimation Algorithm (LACE) is designed. LACE is based on a Scenario-Aware Hyper-Network (SAH-Net) that incorporates the embedding loss to make the Convolutional Neural Network (CNN) based encoder learn to extract the statistical information from the time-space two dimensional features of the CSI. It is combined with a Multi-Layer Perceptron (MLP) based decoder to map the extracted information to the adjusting weights. Our learning design is complemented with analysis to verify that the theoretical performance of LACE is strictly superior to that of the mix-training method, which involves conventionally training the deep network-based channel estimation method using samples from all scenarios. Our results show that in the simulation with finite scenarios, the performance of LACE is comparable to that of the deep network-based channel estimation method trained in each scenario, while having lower complexity. And in the simulation with more realistic infinite scenarios, the performance of LACE is demonstrated to be superior to that of the mix-training method in all test scenarios.

Index Terms—Scenario-Adaptive Channel Estimation, Deep Learning, Neural Network, Scenario-Aware Hyper-Network.

I. INTRODUCTION

With the increase of carrier frequency and bandwidth, the new generation wireless communication system is more sensitive to the communication environments [1]–[3]. The characteristics of the channel state information (CSI) change a lot from one communication environment to another. To ensure reliable and efficient communication, various scenarios and associated channel models have been established to cater to different communication environments [4]. However, the algorithms in these systems must be manually adjusted for each scenario, with the channel estimation module of the receiver being a critical component. The accuracy of CSI estimation is essential for the performance of other modules in the system such as signal detection [5], [6], CSI feedback [7], [8], CSI prediction [9], [10] and beam-forming [11]–[13]. This paper focuses on this scenario-adaptive channel estimation problem, where CSI estimation is tailored to the different scenarios, to address the challenge of varying characteristics of CSIs in different scenarios.

A. Related Work

Classical CSI estimation techniques are based on the theoretical formulation, such as the Least Square (LS) method, Minimum Mean Square Error (MMSE) method, Linear Minimum Mean Square Error (LMMSE) method and Fourier Transform (FT) method [14], which cannot address the scenario-adaptive channel estimation problem. The estimation of LS method is not accurate enough when the Signal to Noise Ratio (SNR) is low, because the statistical CSI information is not considered. MMSE method is designed based on the accurate probability density function of the CSIs, which is challenging to obtain for each scenario in real applications. As an approximation, LMMSE method is proposed under the assumption of linear operation. The covariance matrices of the CSIs calculated from numerous CSI samples in each scenario are required in LMMSE. FT method is mostly used in Orthogonal Frequency Division Multiple Access (OFDM) system. The smoothing filter is set in this method to filter the noise and recover the CSI. The problem is that the parameters of the smoothing filter are set manually for each scenario. Meanwhile, the estimation is also not accurate enough facing with low SNR case for the unclear delay information. As the number of antennas and the bandwidth increases, sparsity of CSI is observed in delay and angular domains. So the channel estimation problem can be modeled as a Compressed Sensing (CS) problem [15]. With the prior information of sparsity, some iterative algorithms have been proposed to obtain the estimated CSI, such as Iterative Soft Threshold Algorithm (ISTA), Approximated Message Passing (AMP) algorithm and Online Approximated Message Passing algorithm (OAMP) [16]–[18]. However, these methods achieve satisfactory performance for CSIs under specific sparsity structures, and may not generalize well to diverse scenarios.

Deep Learning (DL) based approaches have been recently introduced for channel estimation. The researches are mainly divided into two categories: model-driven and data-driven. For model driven DL methods, the channel estimation problems are still modeled as CS problems. Some operations in the traditional iterative framework are replaced by deep neural networks, or some parameters are set as learnable [19]. For data driven methods, the channel estimation problems are
seen as image de-noising and super resolution problems. Convolutional Neural Networks (CNN) are utilized to recover the finer-grained feature of the CSIs [20]–[22]. However, these DL methods can’t perform well when faced with the changing transmission scenarios. Once training is done, the parameters of the network are fixed and can only deal with the CSIs with the same statistical distribution as the training CSIs. The neural network trained in one scenario can not be applied to another scenario directly.

Aiming at the scenario-adaptive channel estimation problem, there are two ways that have been considered in the literature. One of them is online learning [23] and the other one is dynamic network. Firstly considering the idea of online learning, SwitchNet is proposed in [24]. In SwitchNet, several networks are pre-trained well for channel estimation and signal detection in different scenarios with CSIs of different delay spreads. A flexible learnable scalar is designed to make these networks be chosen adaptively, which is trained online with a few pilots. The performance can be guaranteed by SwitchNet in different scenarios with CSIs of different delay spreads. However, delay spread is just one of the key parameters of CSI. SwitchNet may not perform well if more than one parameters of CSIs are changed in different scenarios. In [25], Model-Agnostic Meta Learning (MAML) technique is used. Channel estimation and signal detection are considered together and the pilots are used to fine-tune the well initialized network in each scenario. The meta trained initialization can be seen as the average of different scenarios. So this method can not guarantee the optimal performance in each scenario.

Dynamic network architectures are more suitable to realize scenario-adaptive channel estimation [26]. Dynamic network is a network with dynamic parameters or architectures. If a well-trained dynamic network is deployed and its parameters or architectures can always be changed to the optimal with different input instances, the problem could be well addressed. One way to realize instance-wise dynamic network is hyper-network proposed in [27]. Hyper-network is a technique that uses a neural network to produce parameters for another neural network. It has recently found wide use in the deep learning and meta-learning fields [28]–[35], such as language modeling [36], 3-D shape representation [37], [38], continual learning [39], hyper-parameter setting [40], one-shot learning [41], [42], neural architecture search [43], [44] and Bayesian learning [45], etc. Once the training is finished, the parameters of the hyper-network are fixed, while those of the main network are not. They are produced by the hyper-network and changed with its inputs. Relevant to scenario-adaptive CSI estimation, in [46], Hyper-Dual-CNN is proposed for channel estimation to improve the robustness of the proposed Dual-CNN. In Hyper-Dual-CNN, several networks are pre-trained well in different scenarios first and then the hyper-network is trained to learn to output a scalar to combine the pre-trained networks, which makes the performance better in different scenarios. The problem is that with the increase of the number of scenarios, a large amount of networks need to be trained well and stored, which greatly impacts the training latency and memory consumption. In [47], the hyper-network is added to an unfolding algorithm for the adaptive channel estimation problem. In this method, the number of paths of CSI as well as the SNR are considered as the input of hyper-network and a key factor of the unfolding algorithm is generated by the hyper-network. The performance is evaluated to exceed that of the mix-training unfolding algorithm, but the requirements of the prior information, which are the number of paths and SNRs can not be obtained easily in real application.

### B. Main Contributions

The above techniques have limited practicality in the coming generations of networks, with the rapidly growing diversity of scenarios. The key statistical parameters of CSIs of different scenarios could be different, such as Delay Spread (DS), Angular Spread (AS), number of paths, transmitting and receiving angles, etc. This leads to a big difference among the characteristics of CSIs in different scenarios and brings great challenge to the design of a scenario-adaptive channel estimation method.

In this paper, we propose to solve this challenging problem through extracting the prior statistical information of CSI and adjusting the channel estimation method according to the extracted information automatically by neural networks. Specifically, a Learning-Based Scenario-Adaptive Channel Estimation Algorithm (LACE) is proposed based on this idea, where Scenario-Aware Hyper-Network (SAH-Net) is designed in LACE to utilize the extracting and adjusting process based on hyper-network. The contributions of this paper can be summarized as follows:

- We summarize the general pipeline for scenario-adaptive channel estimation problems based on the previous researches. And we propose LACE to solve this problem through realizing the pipeline with deep neural networks.
- We propose SAH-Net in LACE to realize the extracting and adjusting process. Considering the time-space feature hidden in the CSI, we design CNN based encoder in SAH-Net and develop embedding loss to lead the encoder to extract the characteristics of CSI. MLP based decoder is designed to adjust the convolutional filters of the backbone channel estimation method, employing a Super Resolution Convolutional Neural Network (SRCNN), based on the extracted embeddings for each scenario.
- The performance of LACE is analyzed theoretically. It is found that the output of LACE can be theoretically modeled as Mixture of Gaussian (MoG) distribution, which is the same as that of the real posterior distribution of CSIs in the scenario-adaptive channel estimation distribution. It is further proved that the performance of LACE is better than that of mix-training SRCNN in the scenario-adaptive channel estimation problem.
- We conduct extensive experiments in both limited multiple scenarios and more realistic infinite scenarios, demonstrating the superiority of LACE. Ablation study is also conducted to confirm the effectiveness of SAH-Net and the specially designed embedding loss.

Instead of utilizing hyper-network to learn selecting one scalar based on several pre-trained deep networks, or one key parameter in the unfolding algorithm, the SAH-Net learns the
information it needs to extract from the received pilots and how to adjust the parameters of the channel estimation method to the optimal in each scenario directly. We evaluate our LACE on the scenario-adaptive channel estimation problem in multiple finite scenarios and more realistic infinite scenarios respectively. The results show that LACE achieves better performance than mix-training SRCNN in arbitrary scenario. Even more, LACE can achieve nearly the optimal performance in each scenario in the simulation of multiple finite scenarios. Ablation studies confirm the effectiveness of SAH-Net and the specially designed embedding loss.

The rest of this paper is organized as follows. In Section II, we will describe the system models of the scenario-adaptive channel estimation. The general pipeline and the motivation of LACE will be clarified in Section III. In Section IV, the details of LACE will be elaborated. And the theoretical analysis of LACE will be conducted in Section V. Simulation results will be presented in Section VI. Finally in Section VII, we will conclude the whole paper.

II. SYSTEM MODEL OF SCENARIO-ADAPTIVE CHANNEL ESTIMATION

In this section, the system model of scenario-adaptive channel estimation is described, which include the model of wireless channels, the definition of scenarios and the scenario-adaptive channel estimation problem.

A. Wireless Channel Models and Definition of Scenarios

The model of wireless channels in this paper follows the definition in 3GPP TR 38.901 [4]. As mentioned in [4], the CSI is modeled through a combination of slow fading and fast fading. The slow fading is determined by the parameters such as path loss, line of sight probability and shadow fading. The fast fading is determined by large and small scale parameters. The large scale parameters mainly include delay spread (DS) and angular spreads (AS). The small scale parameters contain the delays, cluster powers, arrival and departure angles for both azimuth and elevation and the cross polarization power ratios (XPR). All these parameters are gathered together in matrix $\Phi$ in this paper. From the statistical perspective, the distribution of the CSI is determined by the parameters $\Phi$.

In practical applications, the CSIs in a given environment are typically characterized by similar statistical distributions with similar parameters. Therefore, for the purpose of defining scenarios, we assume that the CSIs in a given scenario follow the same statistical distribution in this paper. This definition implies that the number of scenarios in real applications is effectively infinite. Several classic scenarios have been defined with the Clustered Delay Line (CDL) channel model in [4], which are respectively CDL-A, CDL-B, CDL-C, CDL-D and CDL-E. The former three scenarios are the classic scenarios with None-Line-of-Sight (NLOS) channel and the last two scenarios are with Line-of-Sight (LOS) channel. The parameters are fixed in these scenarios and differ from each other.

B. Scenario-Adaptive Channel Estimation

The scenario-adaptive channel estimation problem involves estimating the CSIs for all possible scenarios. The objective is to perform efficient channel estimation for any scenario encountered. We consider OFDM uplink channel estimation where it is assumed there are $M$ different scenarios in the scenario-adaptive channel estimation problem. The distribution of the CSI in the $i$-th scenario is determined by the parameters $\Phi_i$. Only the statistical parameters are considered to be different for CSIs in different scenarios in this paper. The other settings are considered to be the same. In each scenario, there is a base station (BS) with $N_i$ antennas serving a number of $N_i$-antennas users, each in $N$ orthogonal sub-carriers. Considering a Transmission Time Interval (TTI) with $T$ OFDM symbols, it is assumed that only the $\lfloor T/2 \rfloor$-th symbol is selected as the pilot symbol. As for the pilot symbol, different sub-carriers are inserted with pilots of different transmitting antennas evenly. So the channel estimation of different transmitting antennas are separated through sub-carriers in OFDM systems. For this reason, the MIMO channel estimation problem can be seen as multiple independent SIMO channel estimation problems. The number of sub-carriers used for pilots of each transmitting antenna is $N_p = \frac{N}{N_i}$. Considering the first transmitting antenna of one user for simplicity, the transmitted pilots in the $i$-th scenario is denoted as $s_{i,p} \in C^{N_p \times 1}$. The received pilots can be obtained as

$$Y_{i,p} = [y_{i,p}^1, \ldots, y_{i,p}^{N_p}] = \text{diag}(s_{i,p})[h_{1,i,p}^1, \ldots, h_{i,i,p}^{N_p}] + [n_{i,p}^1, \ldots, n_{i,p}^{N_p}] = \text{diag}(s_{i,p})H_{i,p} + N_{i,p},$$

where $Y_{i,p} \in C^{N_p \times N_r}$ denotes the received pilots in the $i$-th scenario. $y_{k,i,p}^r \in C$ is the $k$-th row of $Y_{i,p}$, indicating the received pilots of the $k$-th receiving antenna. $diag(s_{i,p}) \in C^{N_p \times N_p}$ denotes the matrix with the diagonal elements as $s_{i,p}$. $H_{i,p} \in C^{N_p \times N_r}$ is the frequency and antenna domain CSI of the pilot sub-carriers in the $i$-th scenario and $h_{k,i,p}^r \in C^{N_r \times 1}$ is its $k$-th row. $N_{i,p} \in C^{N_p \times N_r}$ denotes the white Gaussian noise at the receiver side, i.e.

$$N_{i,p} \sim CN(0, \sigma^2 I_{N_p \times N_r}),$$

in which $\sigma^2$ denotes the variance of the complex Gaussian distribution and it indicates the noise level at the receiver side.

In the scenario-adaptive channel estimation problem, the $M$ scenarios defined above are considered. Our goal is to design an channel estimation method $f(\cdot)$ efficient for arbitrary one of the $M$ scenarios. $f(\cdot)$ is the mapping from the received pilots to the estimated CSIs. So the scenario-adaptive channel estimation can be formulated as the following optimization problem

$$\min_{f} \| f(Y_{i,p}) - H_{i,p} \|^2_F, 1 \leq i \leq M,$$

where $\| : \|_F$ denotes the Frobenius norm. $H_{i,p} \in C^{N \times N_r}$ denotes the frequency and antenna domain CSI of all sub-carriers in the $i$-th scenario. The optimization problem consists
of $M$ sub-problems, each denoting the channel estimation problem in one scenario. The optimal solution of equation (3) can produce optimal estimation of CSIs in all $M$ scenarios.

III. MOTIVATIONS OF SCENARIO-ADAPTIVE CHANNEL ESTIMATION METHOD

In this section, the general pipeline for the scenario-adaptive channel estimation problem is summarized and the motivations of the proposed scenario-adaptive channel estimation method will be elaborated.

A. General Pipeline

Aiming at the optimization problem in equation (3), the existing algorithms can solve it theoretically, such as MMSE, LMMSE, FT and deep learning based methods. These algorithms have the same feature which is requiring the prior statistical information of the CSI in each scenario. Taking the general $i$-th scenario for example, they can all be included in the same pipeline, which is shown as

$$\hat{H}_i = f_w(\Phi_i)(Y_{i,p}) = f_w(g(Y_{i,p}))(Y_{i,p}),$$

where $\hat{H}_i \in \mathbb{C}^{N_r \times N_t}$ is the estimation of the full CSI $H_i$ in the $i$-th scenario. $f_w(\cdot)$ denotes the channel estimation method determined by the parameters $w(\cdot)$. $w$ is a function mapping the received pilots to the statistical information of the CSI.

In this pipeline, the prior statistical information of CSI is used to form the parameters of the channel estimation algorithm. And then the algorithm is utilized to estimate the CSI from the received pilots. However, the accurate statistical information of CSI can hardly be obtained due to the fast changing property of the communication environment. It should be extracted or calculated with the historical and current received pilots. And the extracted prior information is mapped to the parameters of the channel estimation method. The pipeline is concluded in Fig. 1.

Fig. 1: The general pipeline for scenario-adaptive channel estimation.

If the statistical information of CSI in the current $i$-th scenario is perfectly known, the general pipeline turns into the ideal MMSE estimation method, which is the closed-form optimal solution of the optimization problem in equation (3). The prior probabilistic density function can be used as the parameters of the MMSE channel estimation method and it can be written as

$$w(\Phi_i) = p(H_i; \Phi_i).$$

With this prior information, the posterior probabilistic density function $p(H_i | Y_{i,p}; \Phi_i)$ can be calculated and further the ideal MMSE estimation of the problem can be obtained as

$$f_w(\Phi_i)(Y_{i,p}) = \mathbb{E}[H_i | Y_{i,p}; \Phi_i],$$

where $\mathbb{E}[\cdot]$ denotes the expectations.

If the probability density function of CSI can not be obtained and only the covariance matrix of the CSI $R_{H_i, H_i} \in \mathbb{C}^{N_r \times N_t}$ is known and written as

$$w(\Phi_i) = R_{H_i, H_i},$$

In this way, the pipeline turns into the LMMSE method, and the estimated CSI can be expressed as

$$f_{w_i}(\Phi_i)(Y_{i,p}) = R_{H_i, H_i,p}(R_{H_i, H_i,p} + \sigma^2(diag(s_{i,p})diag(s_{i,p})^{-1}))^{-1} \times diag(s_{i,p})^T Y_{i,p},$$

where $R_{H_i, H_i,p} \in \mathbb{C}^{N_r \times N_t}$ denotes the covariance matrix between the full CSI and the CSI on the pilot sub-carriers. It is a part of $R_{H_i, H_i, H_i} \in \mathbb{C}^{N_r \times N_t}$ which is the self-covariance matrix of the CSI on the pilot sub-carriers, which is a part of $R_{H_i, H_i, H_i}$. $(\cdot)^+$ denotes the pseudo-inverse operation.

If the covariance matrix is also unknown and only the delay information of the CSI can be obtained and denoted as $D_i \in \mathbb{C}^{N_r \times N_t}$, which is a diagonal matrix. At the positions corresponding to the delay information on the diagonal, each diagonal element takes the value of 1, while all other elements are set to 0. It can be seen as the parameters of the FT method as

$$w(\Phi_i) = D_i.$$  

So the pipeline can turn into the FT method and the estimated CSI can be expressed as

$$f_{w_i}(\Phi_i)(Y_{i,p}) = F D_i F^H Q diag(s_{i,p})^T Y_{i,p},$$

where $F \in \mathbb{C}^{N_r \times N_t}$ denotes the Fourier transform matrix and $(\cdot)^H$ is the conjugate transpose operation. $(\cdot)^+$ denotes the pseudo-inverse operation. And $Q \in \mathbb{C}^{N_r \times N_t}$ is the interpolation matrix. Different values of $Q$ denote different interpolation methods, such as nearest interpolation, linear interpolation and quadratic interpolation, etc.

For the AI based methods, they can also be seen as obeying the pipeline. The training data-set in each scenario is considered as the prior information, which is hidden in the large amount of labeled data. The parameters of the algorithm are determined through training with the data-set in each scenario. In real application, the received pilots are firstly classified to one scenario, and the well-trained method in this scenario is utilized to estimate the CSI.
From the above analysis, it can be seen that all the methods to solve the problem in equation (3) can be included in the general pipeline presented in equation (4). To solve the challenging scenario-adaptive channel estimation problem, the prior information of the current scenario should be obtained first and the parameters of the channel estimation method should be adjusted or formed by the prior information. In different scenarios, the parameters of the method should be different. In previous methods, the statistical information extracting and the parameters forming are both realized manually. However, considering that the communication environment changes quickly, it is not practical in real application to calculate and store the prior information of the CSI. The prior information can only be extracted with the current or very recent past data. With these limitations, the performance of previous channel estimation methods can not be guaranteed.

B. Idea of the Proposed Method

One way to solve the above problem is to realize the above pipeline through the way of learning, which is exactly the idea of LACE. The $f$, $g$ and $w$ in equation (4) are all modeled as neural networks in LACE. After training, LACE is expected to learn to extract the statistical information from the current received pilots and produce the adjusting weights to adjust the parameters of the channel estimation network to the optimal. The whole process is trained well together with a well designed mixed training data-set off-line. The data-set contains various of received pilots, true CSIs as well as the statistical information containing all kinds of scenarios without any online training or fine-tuning.

IV. Network Design

Based on the above idea, the detail of LACE will be elaborated in this section. In LACE, SRCNN is used as the backbone network for channel estimation. Based on the backbone, we propose a SAH-Net based on the hyper-network to extract the low dimensional embeddings of the CSI after training. The two networks are all based on CNN structure to deal with the time-space feature of the CSI. The decoder is realized through a multi-layer perceptron (MLP) to map the extracted embeddings to the weights for adjustment. The approximation property is considered for utilizing the structure of MLP as the decoder.

The second key point of LACE is its training method. The training data-set must include numerous training CSI samples from all possible scenarios. Adjustment part and channel estimation part are both conducted and the parameters of the SAH-Net and the backbone SRCNN are optimized together in a training step using the finally mean square error (MSE) loss between the estimated CSI and the ground truth. Embedding loss is also designed as the regularization term to ensure the SAH-Net has the ability to extract the accurate statistical information of the CSI in the current scenario.

The details of the backbone method SRCNN, the structure design of SAH-Net and the training method of LACE will be presented in the following subsection.

A. SRCNN for Channel Estimation

Generally, the channel estimation of OFDM system can be considered as a super resolution and de-noising problem. Take the $i$-th scenario for example. Firstly, the LS estimation of the CSI of the pilot sub-carriers can be obtained as

$$H_{ls}^i = \text{diag}(s_{i,p})^{-1}Y_{i,p},$$

(11)

where $H_{ls}^i \in \mathbb{C}^{N_p \times N_r}$ is the estimated CSI of the pilot sub-carriers. Take the expression of $Y_{i,p}$ in equation (1) to the above equation and we can get

$$H_{ls}^i = H_{i,p} + \text{diag}(s_{i,p})^{-1}N_{i,p},$$

(12)

where $N_{i,p}$ still follows complex Gaussian distribution.

In this way, the estimated $H_{ls}^i$ can be viewed as a low-resolution frequency and antenna domain noising CSI. The low resolution CSI is interpolated to high resolution using traditional interpolation methods such as linear interpolation, the nearest interpolation or other high-order interpolation methods. And the estimated frequency and antenna domain noising CSI of all sub-carriers $H_{ls}^i \in \mathbb{C}^{N \times N_r}$ is obtained. Conduct an Inverse Discrete Fourier Transformation (IDFT) in the frequency domain of $H_{ls}^i$ and a Discrete Fourier Transformation (DFT) in its antenna domain, The high resolution delay and beam domain noising CSI $H_{ls}^{1s,d} \in \mathbb{C}^{N \times N_r}$ can be obtained. After this, the following process of the estimation problem is transferred to a pure de-noising problem. The deep learning based methods are efficient enough to solve this kind of problems. In this paper, the SRCNN is considered, of
which the structure is promoted through replacing the Batch-Norm (BN) layer with the Instance-Norm (IN) considering the characteristics of CSI. The details of the structure of SRCNN is shown in the down left corner of Fig. 2. SRCNN consists of three layers. The first layer contains a $9 \times 9$ convolutional layer and a ReLU activation function. The second layer contains a $5 \times 5$ convolutional layer, an IN layer and a ReLU activation function. And the final layer only contains a $5 \times 5$ convolutional layer. The performance can be guaranteed in the specific training scenario according to the previous work [22]. After enough training with the labeled data in the $i$-th scenario, the more accurate estimation of the high-resolution delay and beam domain noising CSI $\hat{H}_d^i \in \mathbb{C}^{N \times N_r}$ can be gotten as

$$\hat{H}_d^i = f_{\theta_i}(\hat{H}_ls; d_i^i),$$

(13)

where $f_{\theta_i}(\cdot)$ denotes the well trained SRCNN in the $i$-th scenario and $\theta_i^i$ is its parameters after enough training. Finally, the estimated full frequency and antenna domain CSI $\hat{H}_i$ can be obtained through the DFT and IDFT transformation to $\hat{H}_d^i$.

However, the well-trained SRCNN can only guarantee the performance in scenarios of which the CSIs have the same statistical characteristics as the training CSIs. As shown in Fig. 3, the SRCNN trained in CDLA scenario has a poor performance in CDLB scenario, compared with the SRCNN trained exactly by the CSIs from the CDLB scenario. There are two ways to solve this problem intuitively, one is to train a specific SRCNN for each scenario, which is called specific training method. Another one is to train a single SRCNN with mixed CSIs from all scenarios, which is called mix-training method. These two methods are both not good enough in application. To solve this tough problem, we propose the SAH-Net to adjust the parameters of the backbone SRCNN automatically for different scenarios. The detail of the SAH-Net is presented in the following.

**B. Design of Scenario-Aware Hyper-Network**

We design our SAH-Net based on the hyper-network with the structure of auto-encoder. In the encoder part, we stack the preprocessing network and the statistical information extractor to process the pre-estimated CSI and extract the statistical information. Meanwhile in the decoder part, we utilize the two-layers MLP to map the extracted information to the adjusting weights. We further develop an effective adjustment strategy,
which is to adjust the convolutional filter of the SRCNN with the adjusting weights. The details of the structure of these three networks and the adjustment strategy are presented below and shown in Fig. 2.

Preprocessing network. The preprocessing network is used to extract clean CSIs from the noising input CSIs. Another well trained SRCNN with mix-training method is selected as the preprocessing network. The SRCNN takes the LS-estimated noisy CSI as the input and output a cleaner CSI. The cleaner CSI is utilized as the prior information and input to the statistical information extractor. The reason for designing the preprocessing network is that the information of the CSI is totally submerged in the white Gaussian noise in the case of low SNR. No statistical information can be extracted directly using the noisy CSI. So the noise level should be decreased with some method. In this paper, we choose to employ a well trained SRCNN with mix-training method due to its reasonable performance for all scenarios and its ability to decrease the noise level.

Statistical information extractor. The structure of the statistical information extractor is a two-layer CNN with an additional regression layer. Each convolutional layer contains a $3 \times 3$ convolution filter and a ReLU activating function. An average pooling layer follows the above layers and the low dimensional vector can be obtained after the average pooling operation. The regression layer is indeed a fully-connected layer to map the obtained low dimensional vector to the extracted embedding. The reason for using CNN as the statistical information extractor is that the CSI in the domain of angular and delay is a 2-dimensional (2-D) matrix. The values of different positions in the 2-D matrix represent the correlation of the time and space of the CSI, which can be seen as the statistical features of the CSI and extracted by CNN perfectly.

Decoder. The MLP based decoder is to map the extracted embeddings to the weights for adjustment. Its first layer is a fully connected layer with ReLu activating function. And the second layer is only a fully connected layer. The statistical information is input to the decoder and the output is the adjusting weights for the SRCNN. The reason of using a two-layer MLP as the mapping between the statistical information and the adjusting weights is the universal approximation ability of MLP. The two-layer MLP with non-linear activating function can approximate any mapping when the size of network is large enough [48].

Adjustment strategy. In LACE, we propose to adjust the second convolutional layer and the last convolutional layer of SRCNN to adapt to different scenarios. The outputs of the decoder are two vectors with length 2, which is written as $[v_1, v_2]$. $v_1$ is responsible for the adjustment of the second convolutional layer of the SRCNN and $v_2$ is responsible for that of the last convolutional layer. The adjusted parameters will be utilized as the new parameters of the convolutional layer, which are obtained by multiplying the first element of $v_1$ with the original weights of the convolution filters and its second element with the original bias. The same operation is conducted to the last layer with $v_2$. In this way, the parameters of SRCNN are adjusted by the characteristic of the current scenario. And the joint training can lead SAH-Net to learn how to adjust the parameters to the optimal for each scenario.

C. Training and Utilization of LACE

In this subsection, we describe the training and utilization of LACE in details.

Training process. A data-set of training samples from all scenarios $\mathcal{D} = \{(H_{ls}^{i,l}, H_{ls}^{i,l}, \Phi_{i,l})\}_{i=1}^{D}$ is required for training LACE. $M$ is the number of scenarios and $L$ denotes the number of training samples in each scenario. $(H_{ls}^{i,l}, H_{ls}^{i,l}, \Phi_{i,l})$ denote the $l$-th training sample in the $i$-th scenario. The training sample contains the pre-estimated CSI with LS, the true CSI and its statistical parameters. Firstly considering to input the general $k$-th batch of data pairs containing $J$ samples to the SAH-Net, the low dimensional extracted embedding $g(H_{ls}^{j})$ can be obtained, where $j = 1 \ldots J$ and $g$ denotes the mapping of the encoder of SAH-Net. In order to lead $g$ to learn how to extract the statistical information of the CSI, the following loss function for embedding dubbed embedding loss is derived as

$$Loss_{emb} = \sum_{j=1}^{J} \parallel g(H_{ls}^{j}) - \Phi_j \parallel^2 . \quad (14)$$

After this, the extracted embedding vector is input to the decoder of the SAH-Net and the adjusting weights $w(g(H_{ls}^{j}))$ are the output, where $w$ denotes the mapping of the decoder of SAH-Net. The adjusting weights are put to the SRCNN and the adjusted SRCNN $f_{w(g(H_{ls}^{j}))}$ can be obtained. After being adjusted, the pre-estimated CSI with LS is input to the SRCNN and the estimated CSI $f_{w(g(H_{ls}^{j}))}(H_{ls}^{j})$ is the output. Finally, the MSE loss is given by the estimated CSI and the true CSI as follows

$$Loss_{recon} = \sum_{j=1}^{J} \parallel f_{w(g(H_{ls}^{j}))}(H_{ls}^{j}) - H_{ls} \parallel^2 . \quad (15)$$

The final loss function of LACE is the combination of the above two losses, which can be expressed as

$$Loss = Loss_{recon} + \lambda Loss_{emb} , \quad (16)$$

where $\lambda > 0$ is the weight for the embedding loss. It is used to balance the effect of the two loss functions in the training, and it is determined according to the training data in real application. In this paper, we set $\lambda = 1$.

The final stage involves back-propagating the final loss of the LACE to derive the gradients of all trainable parameters and optimize these parameters one step with the adaptive moment estimation (Adam) method. The LACE will have been well trained after repeating the above iteration for several times until convergence.

We exploit flow-data training in the training process of LACE. LACE is expected to learn what to extract from the inputs and how to adjust the SRCNN with the extracted information. So numerous training samples are required by LACE to realize the expectation. Considering the memory constraints, flow-data training is capable of dealing with this...
problem, which is to generate as many as possible training samples according to the capability of the computer storage initially, then re-generate part of the samples after each epoch of training. In this way, the number of samples used for training can be increased greatly.

Utilization process. The LACE can be utilized in real application after it is well trained. It can be used in all scenarios directly without any online training or fine-tuning. In the use process, the received pilot in the specific scenario firstly undergo the LS estimation, interpolation and the domain transformation to obtain the pre-estimated CSI. Then the pre-estimated CSI is input to the SAH-Net and output the adjusting weights specially for the current scenario. The weights are given to the SRCNN to adjust its parameters and the well weights especially for the current scenario. Hence the accurate enough CSI can be estimated finally. The whole process is conducted automatically by deep neural networks without any manual operation.

V. THEORETICAL ANALYSIS

In this section, we analytically prove that LACE is more effective in scenario-adaptive channel estimation than mix-training SRCNN with enough training samples and ideal training process. The conclusion is summarized in Theorem 1 and the materials for proof are presented in Lemma 1, Lemma 2 and Lemma 3. In the theoretical analysis, we consider the channel estimation over a single antenna and sub-carrier for simplicity. It can be extended to the channel estimation of multiple antennas and sub-carriers easily. The CSIs in the arbitrary i-th scenario are assumed to obey the complex Gaussian distribution with the mean \( \mu_i \) and the variance \( \sigma_i \), which is known as the Rayleigh fading channel model. So the statistical parameters of the CSIs in the i-th scenario is \( \Phi_i = (\mu_i, \sigma_i) \). The received signal in the i-th scenario with these assumptions is derived as

\[
y_i = s h_i + n_i, \tag{17}
\]

where \( s \) is the transmitting pilots, \( h_i \) denotes the CSI obeying the Rayleigh fading determined by the parameters \( \Phi_i \). It is known that the prior distribution of \( h_i \) is the complex Gaussian distribution with the probability density function denoted as \( p_i(h_i) \). \( y_i \) is the received pilots and \( n_i \) is the white Gaussian noise. It can be derived from equation (17) that the distribution of \( y_i \) given \( h_i \), which is denoted as \( p(y_i|h_i) \) also obeys the complex Gaussian distribution. The posterior distribution of the CSI \( h_i \) given the received pilots \( y_i \) is expressed as \( p(h_i|y_i) \).

Lemma 1. With the assumption of Rayleigh fading channel and Additive White Gaussian Noise (AWGN), the posterior distribution of the CSI \( h_i \) given the received pilots \( y_i \) in the i-th scenario \( p(h_i|y_i) \) obeys complex Gaussian distribution.

Proof: \( p(h_i|y_i) \) can be derived according to the Bayesian formula as

\[
p(h_i|y_i) = \frac{p(y_i|h_i)p(h_i)}{p(y_i)}. \tag{18}
\]

As we know, \( p(y_i|h_i) \) and \( p(h_i) \) are both complex Gaussian distribution according to the assumptions. \( p(y_i) \) is a constant independent of \( h_i \). So \( p(h_i|y_i) \) also obeys the complex Gaussian distribution.

The mean and variance of \( p(h_i|y_i) \) are denoted as \( \mu_{i,y_i} \) and \( \sigma_{i,y_i} \). The SRCNN based channel estimation method is denoted as the \( f_\theta(\cdot) \), and the data-set \( D_i = \{(y_{i,l}, h_{i,l})|_{l=1}^L\} \) denotes the training data-set for training the specific SRCNN in the i-th scenario. The loss function for training \( f_\theta(\cdot) \) is the average square error of all training samples, which is denoted as

\[
Loss = \frac{1}{L} \sum_{l=1}^{L} \| f_\theta(y_{i,l}) - h_{i,l} \|^2. \tag{19}
\]

Assume that the number of training data can be infinite and the training process is ideal. The optimal solution \( \theta_i^* \) can be obtained after training. The statistical clarification of the SRCNN for channel estimation can be given by Lemma 2.

Lemma 2. Assume that \( \theta_i^* \) is the optimal solution of equation (19). Then \( f_{\theta_i^*}(y_i) \) is the maximum-likelihood estimation of the mean \( \mu_{i,y_i} \) of \( p(h_i|y_i) \) with \( L \) observations. And when \( L \) approximates infinite, \( \mathbb{E}_{h_i}[f_{\theta_i^*}(y_i)] \) approaches \( \mu_{i,y_i} \), and \( \mathbb{V}_{h_i}[f_{\theta_i^*}(y_i)] \) approaches 0. \( \mathbb{E}_{h_i}[] \) and \( \mathbb{V}_{h_i}[] \) represent the mean and variance of the random variable.

Proof: Considering the assumption that there are many enough training data, let’s assume \( y_i \) is fixed for these \( L \) training samples \( \{y_i, h_i\}_{l=1}^L \). \( f_{\theta_i^*}(y_i) \) can be seen as the optimal solution of the loss function, which can be derived as

\[
f_{\theta_i^*}(y_i) = \arg\min_{f_\theta} \frac{1}{L} \sum_{l=1}^{L} \| f_\theta(y_{i,l}) - h_{i,l} \|^2 = \arg\max_{f_\theta} \frac{1}{2\pi \sigma_{i,y_i}^2} \sum_{l=1}^{L} (h_{i,l} - f_\theta(y_{i,l}))^2 = \arg\max_{f_\theta} \prod_{l=1}^{L} p(h_{i,l}|y_{i,l}; f_\theta(y_{i,l}), \sigma_{i,y_i}^2) \tag{20}
\]

where \( \sigma_{i,y_i}^2 \) is assumed as a fixed finite value. \( p(h_{i,l}|y_{i,l}; f_\theta(y_{i,l}), \sigma_{i,y_i}^2) \) is the likelihood function of Gaussian distribution with mean \( f_\theta(y_{i,l}) \) and variance \( \sigma_{i,y_i}^2 \). \( p(h_{i,l}|y_{i,l}) \) is also Gaussian distribution with mean \( \mu_{i,y_i} \) and variance \( \sigma_{i,y_i}^2 \), according to Lemma 1. Meanwhile the training samples are also drawn from \( p(h_{i,l}|y_{i,l}) \). From the above results, it can be seen clearly that the optimization of the loss function is indeed obtaining the maximum-likelihood estimation of the mean \( \mu_{i,y_i} \) of \( p(h_{i,l}|y_{i,l}) \) with its L observations. And \( f_{\theta_i^*}(y_i) \) is the optimal solution.

According to [49], the optimal solution of the maximum-likelihood estimation can approach the minimum variance unbiased (MVU) estimation and reach the Cramer-Rao lower bound (CRLB) when \( L \) extends infinite. So \( \mathbb{E}_{h_i}[f_{\theta_i^*}(y_i)] \) approximate \( \mu_{i,y_i} \) and the variance of the estimation is exactly
the CRLB which is derived as
\[
\mathbb{V}_{h_i}[f_{\theta_i^t}(y_i)] = \frac{1}{-\mathbb{E}_{h_i}\left[\frac{\partial^2}{\partial \mu^2} \prod_{i=1}^{L} p(h_i, y_i ; \mu, \sigma^2_{y_i})\right]} = \frac{\sigma^2_{y_i}}{L},
\] (21)
where \(\mu\) is an assuming variable. It can be seen from the above derivation that \(\mathbb{V}_{h_i}[f_{\theta_i^t}(y_i)]\) approaches 0 when \(L\) intends infinite.

**Lemma 3.** \(\theta_i^t\) is assumed as the optimal solution of equation (19) with many enough training samples and ideal training process. The mean square error (MSE) of the SRCNN-based channel estimation method \(f_{\theta_i^t}(\cdot)\) with parameters \(\theta_i^t\) of the \(i\)-th scenarios is
\[
MSE(f_{\theta_i^t}(\cdot)) = \mathbb{E}_{y_i}[(\sigma^2_{y_i})].
\] (22)

**Proof:** Firstly, the mean square error (MSE) under consideration is defined as
\[
MSE(\hat{h}) = \int_y \int \parallel h - \hat{h} \parallel^2 p(h, y)dhdy,
\] (23)
where \(\hat{h}\) is the estimated CSI. \(y\) and \(h\) are assuming variables denoting received pilots and the true CSI. So the mean square error of \(f_{\theta_i^t}(y_i)\) can be derived according to Lemma 2 and the above definition as
\[
\begin{aligned}
MSE(f_{\theta_i^t}(\cdot)) &= \int_{y_i} \int_{h_i} \parallel h_i - f_{\theta_i^t}(y_i) \parallel^2 p(h_i, y_i)dh_idy_i \\
&= \int_{y_i} \int_{h_i} \parallel h_i - f_{\theta_i^t}(y_i) \parallel^2 p(h_i, y_i)dh_i dy_i \\
&= \mathbb{E}_{h_i, y_i}[\parallel h_i - f_{\theta_i^t}(y_i) \parallel^2]dy_i \\
&= \mathbb{E}_{y_i}(\sigma^2_{y_i}),
\end{aligned}
\] (24)

With some conclusions in one specific scenario above, let’s turn to the scenario-adaptive channel estimation problem. Assume there are CSI samples of \(M\) scenarios and \(L\) samples in each scenario available for training. The training data-set can be represented as \(\mathbb{D} = \bigcup_{i=1}^{M} \mathbb{D}_i\) and \(\mathbb{D}_i\) was defined before.

Due to the fact that the CSIs in different scenarios obey the complex Gaussian distribution with different statistical parameters, the prior distribution of the CSIs in mixing scenarios is the Mixture of Gaussian (MoG) distribution, which is expressed as
\[
p(h) = \sum_{i=1}^{M} \pi_i p_i(h),
\] (25)
where \(\pi_i\) denotes the weights of the \(i\)-th Gaussian component. It is set as \(1/M\) in this section, which inherits the assumption that there are the same amount of samples for each scenario. \(p_i(h)\) is the probability density function of the prior distribution of the CSI in the \(i\)-th scenario. In this way, the posterior distribution of the CSI given the received pilots in the mixing scenarios is also a MoG distribution. The proof is similar with that in Lemma 1. The mean and variance of the posterior distribution can be derived as
\[
\begin{aligned}
\mu_{mix} &= \sum_{i=1}^{M} \pi_i \mu_i, \\
\sigma^2_{mix} &= \sum_{i=1}^{M} \pi_i (\mu^2_i + \sigma^2_{y_i}) - \mu_{mix}^2.
\end{aligned}
\] (26)

**Theorem 1.** In the scenario-adaptive channel estimation problem for \(M\) scenarios with \(L\) training CSIs in each scenario and \(L\) is assumed to approach infinite. The training is assumed to be conducted ideally. The ideal solution of mix-training SRCNN method is denoted as \(f_{\theta_i^m}(\cdot)\) and that of LACE is \(f_{w_{\phi^e}(\phi^w, \cdot)}(\cdot)\). Then the following two consequences hold and expressed as
\[
\begin{aligned}
g_{\phi^e}(y_i) &= \Phi_i, \\
g_{\phi^w}(y_i) &= \Phi_i.
\end{aligned}
\] (27)

**Proof:** Firstly considering LACE, the encoder of the SAH-Net \(g_{\phi}\) is well trained with embedding loss. So the output of \(g_{\psi}\) is the optimal maximum likelihood estimation of \(\Phi_i\) in the \(i\)-th scenario, which is similar as the conclusion in Lemma 2. With large enough \(L\) and the ideal training process, \(g_{\phi}(y_i)\) is the MVU estimation of \(\Phi_i\) and the variance is the CRLB. And the variance tends to zero when \(L\) tends infinite. So the following formula holds with the assumption of infinite training samples and ideal training process.
\[
g_{\phi^e}(y) = \Phi_i.
\] (29)

With the above consequence, the output of LACE can be seen as following the complex Gaussian distribution with \(f_{w_{\phi^e}(\Phi_i)}(\cdot)\) as the mean. For different \(y\)s from different scenarios, the output of the LACE can be seen as following different complex Gaussian distributions with different means. Indeed, it can be concluded that LACE model the posterior distribution of the CSIs in mixing scenarios as the MoG distribution, which is exactly the true format of the real posterior distribution in the scenario-adaptive channel estimation problem. Due to the assumption that LACE can be trained very well so that each \(f_{w_{\phi^e}(\Phi_i)}(\cdot)\) can fully approximate the mean of the posterior distribution in that scenario according to Lemma 2. The MSE of LACE in the \(i\)-th single scenario can be derived as \(\mathbb{E}_{y}[(\sigma^2_{y_i})]\) according to Lemma 3. And further the MSE in mixing scenarios can be derived as
\[
MSE(f_{w_{\phi^e}(\Phi_i)}(\cdot)) = \mathbb{E}_{y}\left(\sum_{i=1}^{M} \pi_i \sigma^2_{y_i}\right).
\] (30)

Then considering mix-training SRCNN, the output of the mix-training SRCNN can be seen as the estimation of the mean of the posterior distribution of the CSIs in mixing scenarios according to Lemma 2. It is assumed that the training is conducted fully in the ideal case. The output of the mix-training SRCNN can be seen exactly the mean of the posterior MoG distribution of the CSIs. And the estimation error can expressed according to Lemma 3 as
\[
MSE(f_{\theta_i^m}(\cdot)) = \mathbb{E}_{y}(\sigma^2_{mix}).
\]
\[ \mathbb{E}_y \left( \sum_{i=1}^{M} \pi_i (\mu_i^{2, y_i} + \sigma_i^{2, y_i}) - \mu_{mix}^{2} \right), \quad (31) \]

where \( y \) denotes the received pilots in the mixing scenarios.

Under the assumption of ideal training, the performance of mix-training SRCNN and LACE have been obtained, which are in the format of MSE. Further, the comparison of these two MSEs will be conducted. Firstly according to the assumption that \( \pi_i = 1/M \), it can be derived that

\[
\begin{align*}
\mu_{mix}^{2} &= \left( \sum_{i=1}^{M} \pi_i \mu_i^{2, y_i} \right)^2 \\
&= \frac{\sum_{i=1}^{M} \mu_i^{2, y_i} + 2 \sum_{i<j} \mu_i^{2, y_i}, \mu_j^{2, y_j}}{M^2} \\
&\leq \frac{M \sum_{i=1}^{M} \mu_i^{2, y_i}}{M^2} \\
&= \frac{\sum_{i=1}^{M} \mu_i^{2, y_i}}{M} \\
\end{align*}
\]

(32)

After this derivation, it can be further expressed according to (28), (30) and (31) as

\[
\begin{align*}
MSE(f_{\theta_\ast}(.)) - MSE(f_{\theta_\ast, \phi, (y, \cdot)}(.)) &= \mathbb{E}_y \left( \sum_{i=1}^{M} \pi_i \mu_i^{2, y_i} \right)^2 - \mu_{mix}^{2} \\
&= \frac{\sum_{i=1}^{M} \mu_i^{2, y_i}}{M} - \mu_{mix}^{2} \\
&\geq 0.
\end{align*}
\]

From Theorem I, it is proved that the MSE of LACE is smaller than or equal to that of the mix-training SRCNN, which indicate that LACE is more effective in the scenario-adaptive channel estimation problem. Specifically, the encoder of the SAH-Net is proved to have the ability to output the approximated statistical information of CSI in each scenario when fully trained with the embedding loss. And the adjustment with the extracted embeddings makes LACE model the posterior distribution of the CSI as MoG, which is closer to the true posterior distribution format. So LACE is superior than mix-training SRCNN in the scenario-adaptive channel estimation problem. The ability of extracting the statistical information and the superiority than mix-training SRCNN of LACE are also evaluated in the simulation part.

### TABLE I: Parameter settings for the simulations of CSIs in the infinite scenario case.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Ranges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of clusters</td>
<td>[5, 25]</td>
</tr>
<tr>
<td>Cluster power (dB)</td>
<td>[−30, 0]</td>
</tr>
<tr>
<td>Cluster delay (ns)</td>
<td>[0, 20000]</td>
</tr>
<tr>
<td>Cluster AoD (°)</td>
<td>[−180, 180]</td>
</tr>
<tr>
<td>Cluster AoA (°)</td>
<td>[−180, 180]</td>
</tr>
<tr>
<td>Cluster ZoD (°)</td>
<td>[0, 180]</td>
</tr>
<tr>
<td>Cluster ZoA (°)</td>
<td>[0, 180]</td>
</tr>
<tr>
<td>XPR</td>
<td>10</td>
</tr>
<tr>
<td>CAS</td>
<td>[3, 25]</td>
</tr>
<tr>
<td>Loss Probability</td>
<td>[0, 1]</td>
</tr>
</tbody>
</table>

### VI. SIMULATION RESULTS

In this section, we conduct experiments to evaluate our proposed LACE. Firstly, we test our LACE with the scenario-adaptive channel estimation task in finite scenarios. Then, the simulation in more challenging infinite scenarios is conducted. Finally, the ablation study is considered to verify the effectiveness of the SAH-Net and embedding loss. The simulation settings and the results are elaborated as follows.

#### A. Simulation Setup

**Simulation of system model.** The data of the CSIs is simulated based on Quadriga, which is implemented on Matlab. As for the CSI simulation, the carrier frequency is set as 3.5 GHz. 32 antennas are considered for the BS, which are modeled as square array of 4 × 4, each with 45 degree bi-polarization. The number of antennas of the user is set as 2, and the channel estimation problem is only considered for single antenna. The mobile speed of the user is set as 3 m/s. In the simulation, the channel estimation is conducted once in a TTI, in which 14 OFDM symbols exist. The bandwidth is set as 7.08 MHz. Thus there are 192 sub-carriers in an OFDM symbol. The 1-st and the 11-th symbols are the pilot symbols and they are estimated together in our simulation. As for the noise level in the simulation environment, the challenging low SNR regime [−15, −10] dB is under consideration.

**Compared methods.** There are totally four methods for comparison with LACE in our simulation, which are described below:

- **Specific-scenario SRCNN:** Specific-scenario SRCNN is the baseline SRCNN trained in each scenario with numerous of training samples respectively. It can be seen as the upper bound of the performance in the scenario-adaptive channel estimation problem.
- **Mixing SRCNN:** Mixing SRCNN is the baseline SRCNN trained with mixed training samples from all scenarios. It is considered as the baseline method.
- **Hyper SRCNN:** In the ablation study, Hyper-SRCNN is considered to verify the effectiveness of embedding loss. The structure of Hyper-SRCNN is the same as LACE, the difference is that it is trained without the embedding loss.
- **Enhanced SRCNN:** Furthermore, the enhanced SRCNN is selected as the comparison method to verify the effectiveness of SAH-Net. The enhanced SRCNN is formed by adding more convolutional layers to the baseline SRCNN to make it more powerful. In the simulation, the number of the adding layers is 8. This makes sure the enhanced SRCNN has the same complexity with our LACE.

The specific-scenario SRCNN is trained from scratch in each scenario. The baseline SRCNN, the Hyper-SRCNN and the enhanced SRCNN have the same training data-set and tricks with LACE to guarantee the fairness of comparison.

**Implementation Details.** In the finite scenarios simulation, the scenario-adaptive channel estimation problem in five classical scenarios, including CDLA, CDLB, CDLC, CDLD and CDLE, is considered. The training data-set for LACE consists of samples from all the five scenarios, initially 4000...
samples from each scenario. In the flow-data training trick, 500 samples from each scenario are re-generated in each training epoch. The SNR of each training sample is randomly selected from $[-15, -10]$ dB. In the infinite scenarios simulation, the scenario-adaptive channel estimation problem in infinite continuous scenario is under consideration. The infinite continuous scenario means the mixture of all possible channel scenarios, which is a more realistic simulation setting. The method to realize the infinite continuous scenario is to generate the CSIs with randomly selected parameters within the desirable range. The considered parameters and the corresponding ranges are presented in Table I. Same as the first part, initially 20000 samples are generated for the training data-set and further 2000 samples are generated in each training epoch. The SNR of each training sample is also randomly selected from $[-15, -10]$ dB. The codes of the AI methods are all implemented with Pytorch in Python 3.7. The computing platform is an i5-9400 CPU, a RTX-2080 GPU and RAM 16GB.

### B. Results in Finite Scenarios Case

Considering the scenario-adaptive channel estimation in finite scenarios, the performance of LACE will be presented in this subsection. The test scenarios are respectively CDLA, CDLB, CDLC, CDLD and CDLE. 5000 samples with SNR ranging from $[-15, -10]$ dB of each scenario are generated for test.

**Comparison with baselines.** Firstly, we evaluate our LACE in finite scenarios and compare it with baseline methods. The results are presented in Fig. 4. In these five sub-figures, the red curve with mark ‘o’ represents the performance of the mix-training SRCNN, the black curve represents the performance of specific training SRCNN, and finally the green curve with the mark ‘*’ denotes the results of our LACE. Firstly considering the test results in the CDLA scenario, which is presented in (a), the performance of LACE is nearly the same with the SRCNN trained specifically in the CDLA scenario, exceeds that of the mix-training SRCNN. Same as before in (b) which presents the test results in CDLB scenario, the LACE also obtains nearly the same performance as the SRCNN trained specifically in the CDLB scenario, even better with SNR increases. The results indicate that the LACE can adapt to CDLA and CDLB scenario easily although the training data-set is the mixing data from all scenarios. Further considering (c), in which the results in CDLC scenario are presented, LACE obtains better performance than other two methods, even the SRCNN trained specifically in the CDLC scenario. The reason for this phenomenon is that the CDLC-trained SRCNN only achieved the local optimal solution with the training data in CDLC scenario. The SRCNN in LACE is adjusted to jump out of the local optimum with the assistance of the training data from other scenarios. Next in (d), LACE outperforms the baseline method about 10% and has slight distance from the specific-training SRCNN. Finally in the CDLE scenario shown in (e), the performance of LACE is in the middle of that of the mix-training SRCNN and the specific-training SRCNN. These two results show that LACE can have satisfactory performance in CDLD and CDLE scenario, although the performance does not reach that of the specific training SRCNN. From the above results, it can be concluded that LACE can nearly adapt to every scenario it has seen in the training procedure well.

**Effectiveness of embeddings and adjusting weights.** To discover the effectiveness of the extracted embeddings and the adjusting weights of LACE intuitively, the visualization of the extracting embeddings and generating weights in different test scenarios is presented in Fig. 5. In (a) of this figure, each point represents the extracting embeddings reduced to 3-D. The points in different colors denote the extracting embeddings in different scenarios. 1000 samples are selected for each scenario. It can be seen that each extracting embedding stay together with the embedding from the same scenario, but keep a distance with that from other scenarios. For the embeddings in CDLD and CDLE scenarios, they are distributed together due to that the CSIs in CDLD and CDLE scenarios are both line of sight (LOS) channels, leading to nearly the same distribution. This phenomenon shows that LACE extracts nearly the same embeddings for the CSIs in the same scenario, but different embeddings for those from different scenarios. This indicates that LACE has learned to extract the scenario-specific embeddings for each scenario. Also in (b) of Fig. 5, each point represents the generating weights reduced to 3-D. The points in different colors represent the weights for different scenarios. It can also be seen that the generating weights in one test scenario gather together, and those in different scenarios distribute in different categories. From this result, it can be seen that LACE has learned how to generate specific weights to adjust the channel estimation method for CSIs in different scenarios. In this way, the ability of LACE to extract scenario-specific embeddings and generate effective adjusting weights can be seen intuitively.

**Complexity comparison in finite scenarios.** In finite scenarios case, the bottle neck of specific training SRCNN is the complexity. In our simulation, 5 SRCNNs need to be stored in the BS to solve the scenario-adaptive channel estimation problem. From the above results, LACE can achieve nearly the same performance with the specific training SRCNN. And it will be derived that LACE has less complexity than the specific training SRCNN. Their complexities are calculated as follows. As for the specific SRCNN for each scenario, the complexity of one SRCNN is 99072, the whole complexity of this method is $99072 \times 5 = 495360$ considering that there are five scenarios in our simulation. The complexity of our LACE is calculated as 316352. Clearly, it can be seen that the complexity of LACE is smaller than the specific SRCNN method. So it can be seen that LACE is the more effective method for the scenario-adaptive channel estimation problem from the point view of performance and complexity in finite scenarios case.

### C. Results in Infinite Scenarios Case

After evaluating the performance of LACE on the finite scenarios, its effectiveness in more realistic and complex infinite scenarios case will be shown in this subsection. The specific-training SRCNN is trained in each test scenario. LACE and
(a) Testing results in CDLA scenario.  (b) Testing results in CDLB scenario.  (c) Testing results in CDLC scenario.  (d) Testing results in CDLD scenario.  (e) Testing results in CDLE scenario.

Fig. 4: Performance of LACE in the five test scenarios. The comparison methods are respectively SRCNN trained in the test scenario and the mix-training SRCNN.

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>CDLA SRCNN</th>
<th>Mix-training SRCNN</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-14.5</td>
<td>-14</td>
<td>-13.5</td>
</tr>
<tr>
<td>-13.5</td>
<td>-13</td>
<td>-12.5</td>
<td>-12</td>
</tr>
<tr>
<td>-12</td>
<td>-11.5</td>
<td>-11</td>
<td>-10.5</td>
</tr>
<tr>
<td>-10</td>
<td>-9.5</td>
<td>-9</td>
<td>-8.5</td>
</tr>
<tr>
<td>-8</td>
<td>-7.5</td>
<td>-7</td>
<td>-6.5</td>
</tr>
<tr>
<td>-6</td>
<td>-5.5</td>
<td>-5</td>
<td>-4.5</td>
</tr>
<tr>
<td>-4.5</td>
<td>-4</td>
<td>-3.5</td>
<td>-3</td>
</tr>
<tr>
<td>-3</td>
<td>-2.5</td>
<td>-2</td>
<td>-1.5</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1</td>
<td>-0.5</td>
<td>-0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>CDLB SRCNN</th>
<th>Mix-training SRCNN</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-14.5</td>
<td>-14</td>
<td>-13</td>
</tr>
<tr>
<td>-13</td>
<td>-12.5</td>
<td>-12</td>
<td>-11.5</td>
</tr>
<tr>
<td>-11.5</td>
<td>-11</td>
<td>-10.5</td>
<td>-10</td>
</tr>
<tr>
<td>-10</td>
<td>-9.5</td>
<td>-9</td>
<td>-8</td>
</tr>
<tr>
<td>-8</td>
<td>-7.5</td>
<td>-7</td>
<td>-6</td>
</tr>
<tr>
<td>-6</td>
<td>-5.5</td>
<td>-5</td>
<td>-4</td>
</tr>
<tr>
<td>-4</td>
<td>-3.5</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>-2.5</td>
<td>-2</td>
<td>-1.5</td>
<td>-1</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1</td>
<td>-0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>CDLC SRCNN</th>
<th>Mix-training SRCNN</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-14.5</td>
<td>-14</td>
<td>-13</td>
</tr>
<tr>
<td>-13</td>
<td>-12.5</td>
<td>-12</td>
<td>-11.5</td>
</tr>
<tr>
<td>-11.5</td>
<td>-11</td>
<td>-10.5</td>
<td>-10</td>
</tr>
<tr>
<td>-10</td>
<td>-9.5</td>
<td>-9</td>
<td>-8.5</td>
</tr>
<tr>
<td>-8</td>
<td>-7.5</td>
<td>-7</td>
<td>-6.5</td>
</tr>
<tr>
<td>-6</td>
<td>-5.5</td>
<td>-5</td>
<td>-4.5</td>
</tr>
<tr>
<td>-4.5</td>
<td>-4</td>
<td>-3.5</td>
<td>-3</td>
</tr>
<tr>
<td>-3</td>
<td>-2.5</td>
<td>-2</td>
<td>-1.5</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1</td>
<td>-0.5</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>CDLD SRCNN</th>
<th>Mix-training SRCNN</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-14.5</td>
<td>-14</td>
<td>-13</td>
</tr>
<tr>
<td>-13</td>
<td>-12.5</td>
<td>-12</td>
<td>-11.5</td>
</tr>
<tr>
<td>-11.5</td>
<td>-11</td>
<td>-10.5</td>
<td>-10</td>
</tr>
<tr>
<td>-10</td>
<td>-9.5</td>
<td>-9</td>
<td>-8</td>
</tr>
<tr>
<td>-8</td>
<td>-7.5</td>
<td>-7</td>
<td>-6</td>
</tr>
<tr>
<td>-6</td>
<td>-5.5</td>
<td>-5</td>
<td>-4</td>
</tr>
<tr>
<td>-4.5</td>
<td>-3.5</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>-3</td>
<td>-1.5</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>-1.5</td>
<td>-0.5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SNR (dB)</th>
<th>CDLE SRCNN</th>
<th>Mix-training SRCNN</th>
<th>LACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-15</td>
<td>-14.5</td>
<td>-14</td>
<td>-13</td>
</tr>
<tr>
<td>-13</td>
<td>-12.5</td>
<td>-12</td>
<td>-11.5</td>
</tr>
<tr>
<td>-11.5</td>
<td>-11</td>
<td>-10.5</td>
<td>-10</td>
</tr>
<tr>
<td>-10</td>
<td>-9.5</td>
<td>-9</td>
<td>-8.5</td>
</tr>
<tr>
<td>-8</td>
<td>-7.5</td>
<td>-7</td>
<td>-6.5</td>
</tr>
<tr>
<td>-6</td>
<td>-5.5</td>
<td>-5</td>
<td>-4.5</td>
</tr>
<tr>
<td>-4.5</td>
<td>-4</td>
<td>-3.5</td>
<td>-3</td>
</tr>
<tr>
<td>-3</td>
<td>-2.5</td>
<td>-2</td>
<td>-1.5</td>
</tr>
<tr>
<td>-1.5</td>
<td>-1</td>
<td>-0.5</td>
<td>0</td>
</tr>
<tr>
<td>-0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Comparison with baselines. In these five sub-figures, the red curve with mark ‘o’ represents the performance of the mix-training SRCNN, the black curve represents the performance of specific training SRCNN and finally the green curve with mark '*' denotes the results of our LACE. Firstly considering (a) which represents the test results in CDLA scenario, it can be seen that LACE obtains better performance than mix-training SRCNN in CDLA scenario, and the performance gain become larger with the increase of SNR, even reach 20% at −10 dB SNR. Also in (b), LACE obtains better performance than mix-training SRCNN in CDLB scenario. And the performance even achieve that of the specific training SRCNN when the SNR reaches above −11 dB. In CDLC scenario, of which the results are shown in (c), the performance gain of LACE compared with mix-training SRCNN is a little small with low SNR ranging from [−15, −13.5] dB. When the SNR is above −13 dB, the performance gain between LACE and the mix-training SRCNN becomes larger and larger, and finally achieves about 16% at about −10 dB SNR. Finally considering the test results in CDLD and CDLE scenarios shown in (d) and (e), LACE still obtains better performance than mix-training SRCNN and the performance gain is always about 12.5% at nearly all SNRs. The above results indicate the effectiveness of LACE and the SAH-Net. However, due to that the task of learning to extract the statistical information of the CSI from the infinite continuous scenarios training data-set is very difficult, the performance of LACE can not approach that of the specific training SRCNN except in the CDLB scenario at the mix-training SRCNN are both trained with samples from the infinite continuous scenarios data-set. And CDLA, CDLB, CDLC, CDLD and CDLE are still considered as the test scenarios. Also [−15, −10] dB SNR is considered and 5000 samples are generated in each scenario for test. The test results are presented in Fig. 6.

(a) Visualizations of the embeddings extracted by LACE trained in finite scenarios case.  (b) Visualizations of the generating weights of LACE trained in finite scenarios case.

Fig. 5: The visualizations of the extracting embeddings and the generating weights in the test procedure of LACE trained in finite scenarios case. The two vectors are both reduced to 3 dimensions and presented in the 3-D space.
Fig. 6: Performance of LACE in the five test scenarios in the simulation of infinite scenarios case. The comparison methods are respectively SRCNN trained in the test scenario and the mix-training SRCNN.

(a) Testing results in CDLA scenario.  
(b) Testing results in CDLB scenario.  
(c) Testing results in CDLC scenario.  
(d) Testing results in CDLD scenario.  
(e) Testing results in CDLE scenario.

Effectiveness of embeddings and adjusting weights. Further to show that LACE has learned to extract the specialized features of the CSIs in each scenario, the 3-D visualizations of the extracted embeddings and the generating weights of LACE is obtained and shown in Fig. 7. From (a), it can be seen that the embeddings of the CSIs from five test scenarios are distributed in four different clusters with some overlap. Also in (b), the weights are grouped into four categories roughly, with some overlap between CDLA and CDLC, as well as CDLB and CDLC. The embeddings and generating weights of LACE in CDLD and CDLE scenarios are almost overlapped due to that they are all LoS channels. From these results, it can be concluded that LACE has learned the ability to extract different specialized embeddings and generate effective weights for different scenarios. However, the performance is not good enough compared with the finite scenarios case. The reason is that the task of extracting effective embeddings with training data of continuous infinite scenarios is much more difficult.

D. Ablation Study

In the ablation study, the effectiveness of Scenario-Aware Hyper-Network and the embedding loss will be evaluated. LACE and the comparing methods are all trained with samples from the mixing scenarios of CDLA, CDLB, CDLC, CDLD and CDLE. And they are all test in these five scenarios one by one respectively. The results are presented in Fig. 8.

Effectiveness of Scenario-Aware Hyper-Network. In these five figures, the green curve with mark ‘*’ denotes the result of LACE and the blue curve with mark ‘+’ denotes that of the enhanced SRCNN. In CDLA, CDLB and CDLC scenarios, our LACE outperforms the enhanced SRCNN at least 15%. And in CDLD and CDLE scenarios, their performances are above −11 dB SNR.

Fig. 7: The visualizations of the extracting embeddings and the generating weights in the test procedure of LACE trained in infinite scenarios case. The two vectors are both reduced to 3 dimensions and presented in the 3-D space.
very close. With nearly the same complexity, LACE can adapt to every scenario it has seen, while enhanced SRCNN only performs well in CDLD and CDLE scenario. It can be derived that designing an SAH-Net is more effective than directly enlarging the complexity of the baseline SRCNN. In this way, the effectiveness of SAH-Net can be proved.

**Effectiveness of embedding loss.** Further to evaluate the effectiveness of our *embedding loss*, the Hyper SRCNN is selected to be compared with LACE, and the results is shown in the sky blue curve with mark ‘’. From the five figures, it can be seen that LACE always get better performance than Hyper SRCNN in any test scenario. The use of *embedding loss* brings about 5% performance gain in average for all scenarios. This directly verifies the effectiveness of our *embedding loss*.

**VII. CONCLUSION**

In this paper, scenario-adaptive channel estimation algorithm was investigated. LACE was proposed based on the summarized pipeline to address this problem with the way of learning, where an auto-encoder based hyper-network named SAH-Net was designed to extract the statistical information of CSI and learn to obtain the adjusting weights for the channel estimation network. The MSE loss between the estimated CSI and the ground truth as well as the specially developed *embedding loss* are applied for training the LACE. The theoretical analysis indicates that LACE is more effective in scenario-adaptive channel estimation than mix-training SRCNN. Simulation results show that in the case of finite scenarios, LACE can achieve nearly the optimal performance in each scenario with less complexity. Furthermore, in the more challenging case of infinite scenario, the performance of LACE is superior than that of the mix-training SRCNN. The ablation study also confirms the effectiveness of SAH-Net and the *embedding loss*.

**REFERENCES**


