Automatic High-Performance Neural Network Construction for Channel Estimation in IRS-Aided Communications

Haoqing Shi, *Student Member, IEEE*, Yongming Huang, *Senior Member, IEEE*, Shi Jin, *Fellow, IEEE*, Zheng Wang, *Senior Member, IEEE*, Luxi Yang, *Senior Member, IEEE*

***Abstract*—Accurate channel estimation is an essential prerequisite for achieving significant performance gains in intelligent reflecting surface (IRS)-aided communication systems. Recent studies have shown that deep neural network-based channel estimation holds promise as a competitive alternative to conventional methods. However, existing neural network-based approaches typically involve manual design of network architectures through a trial-and-error process, demanding extensive domain knowledge and human resources. In this paper, we propose an automatic approach to construct a high-performance neural network architecture for channel estimation. Our method, called the channel estimation neural network architecture search (CENAS), utilizes a truncated back-propagation optimization search strategy to explore a neural network tailored for channel estimation. By carefully designing a search space tailored to channel estimation tasks, the automatically constructed network surpasses both conventional and deep learning-based channel estimation algorithms. The convergence of our framework's network construction process is comprehensively analyzed, providing formal evidence of its convergence properties. Additionally, the proposed framework exhibits good generalization and applicability by allowing flexible adjustment of hyperparameters to generate networks with varying scales. Empirical results show the stability and the improved performance of CENAS framework, validating its effectiveness and desirability.**

***Index Terms*—Intelligent reflecting surface, channel estimation, neural network architecture search, deep learning.**

I. Introduction

THE intelligent reflecting surface (IRS) is a novel technique that has been introduced to enhance the coverage and capacity of wireless communication systems with low hardware cost and energy consumption [2]–[5]. Composed of massive passive low-cost elements, IRS can intelligently adjust the electromagnetic propagation environment between user equipments (UE) and access points (AP). Thus, by controlling the amplitude response and phase shift of the passive elements, IRS provides high beamforming gain, allowing wireless signals to be steered in the desired directions [6], [7]. Notably, the IRS's utilization of passive elements for signal reflection, rather than active RF chains, significantly reduces hardware and energy costs compared to the traditional active relays [8]. Meanwhile, the deployment of IRS in simultaneous wireless information and power transfer (SWIPT) and secrecy communications are able to bring significant benefits to wireless communication systems in terms of energy harvesting efficiency, information transmission rate, and security enhancement [9]–[11]. With its cost-effectiveness and flexibility, IRS has advantages of becoming a key enabler for smarter radio environments, meeting the rising demand for higher data rates and increased communication device usage. By jointly optimizing the secondary user (SU) transmit power and IRS reflect beamforming, the proposed IRS-aided cognitive radio system is capable of achieving high rates [12]. In NOMA systems, IRS can enhance the system's capacity and energy efficiency by improving channel gain, increasing the number of supported users, and reducing the transmit power required [13].

However, accurately estimating channel state information (CSI) is crucial for realizing the benefits of IRS technology in wireless networks, yet this remains a challenging task due to the passive nature of IRS elements and the high-dimensional cascaded channel in practice [14]. Different from the traditional systems, it is quite difficult to estimate IRS-UE and IRS-AP channels respectively since the CSI of the cascaded channel is the product of AP-IRS channel and IRS-UE channel [15], where the passive devices are incapable of transmitting, receiving and processing pilot signals. In comparison to the channel in conventional systems, another obstacle in IRS aided systems is the larger dimension of the cascaded channel due to hundreds of passive elements. Consequently, the pilot overhead for channel estimation increases rapidly, thus causing the conventional methods prohibitive [16]. While previous studies mainly focus on perfect CSI, it is highly demanded for efficient channel estimation methods in IRS-aided systems to enhance the practicality and viability of this technology.

Numerous efficient algorithms have been proposed to overcome these obstacles. One such method is a least-squares (LS)-based estimation technique proposed in [17], which aims to obtain an unbiased cascaded channel estimator for single-user

This work was supported in part by the National Natural Science Foundation of China under Grant 62225107 and Grant 62371124, in part by the Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20222001, in part by the Fundamental Research Funds for the Central Universities under Grant 2242022k60002, and in part by the Major Key Project of Peng Cheng Laboratory(PCL). This article has been partially published in the IEEE Wireless Communications and Networking Conference (WCNC), 2023 [1]. *(Corresponding author: Yongming Huang)*.

The authors are with the School of Information Science and Engineering and the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China, and also with the Pervasive Communications Center, Purple Mountain Laboratories, Nanjing 211111, China (e-mail: {shihaoqing619; huangym; jinshi; zheng_wang; lxyang}@seu.edu.cn).

multiple-input single-output (SU-MISO) systems. Besides, a compressive sensing (CS)-based channel estimation scheme is given in [18] using the angular-domain channel sparsity. In addition, the authors in [15] first estimate the full channel state information including all the angle and gain information in the first coherence block, and then only re-estimate the channel gains in the remaining coherence blocks with much lower pilot overhead.

On the other hand, deep learning has gained widespread use in wireless communication due to its ability to directly learn the sophisticated mapping from the received data to the desired output. In [19], the paper adopts a novel approach by treating the channel as a 2D image, utilizing image super-resolution and restoration networks to derive channel estimates from pilot data. In [20], the authors leverage deep convolutional neural networks (CNNs) to enhance channel estimation in millimeter-wave massive MIMO systems. Additionally, in [21], neural networks are harnessed for the joint optimization of pilot design and downlink channel estimation within wideband massive MIMO systems to improve spectral efficiency. Regarding the channel estimation as a denoising problem, several data-driven deep learning (DL)-based methods have shown their strengths in IRS channel estimation. For instance, A convolutional deep residual network called CDRN is proposed in [16] to denoise the noisy pilot-based observations. In [22], the authors introduced a residual dense network structure that exploits the low-rank structure of IRS channels to improve the generalization and fitting capabilities. Besides, a ResU-Net architecture with stacked residual units is employed in [14] to enhance feature extraction for high-dimensional cascaded channel reconstruction. Meanwhile, a self-supervised learning (SSL)-based channel estimation solution is shown in [23], which demonstrates an epoch-wise double descent phenomenon and showcased promising results through extensive numerical simulations.

Despite the competitive performance achieved by these DL-based approaches for the channel estimation problem [14], [16], [22], they still need to manually design the network architectures in a trial-and-error manner, which is fragile to the channel matrix dimension changes and human experiential knowledge. These works utilize neural networks architectures borrowed from the existing successful neural network [24], [25], which may not necessarily be the optimal choice for the specific task of channel estimation. Additionally, the network architecture of the aforementioned DL-based methods is susceptible to various factors, including signal-to-noise ratio (SNR) and input dimensions, which may limit them to consistently provide optimal results.

The escalating number of antennas and the diverse real-world scenarios necessitate the development of more flexible and robust channel estimation algorithms for IRS. Motivated by these challenges, we propose a neural network architecture search method for channel estimation. Neural network architecture search (NAS) is an emerging research field that focuses on the automatic design of high-performance neural architectures, eliminating the need for manual design with domain knowledge [26]. Distinguished from hyperparameter optimization methods [27], [28], which primarily aim to find optimal models within predefined, fixed-length parameter spaces, NAS broadens its horizon to variable-length configurations that determine the structural complexities and connectivity of a neural network [26]. By exploring diverse architectural possibilities, NAS offers a more comprehensive search space, making it a more effective tool for addressing complex problems in machine learning. Recent advancements in network architecture search (NAS) draw inspiration from the concepts of learning to learn [29] and meta-learning [26], [30]. The work presented in [26], [31] employs reinforcement learning to train a controller that evaluates the performance of neural networks, opening up new avenues for NAS research. [30] introduces a model-agnostic optimization algorithm based on meta-learning, which aims to develop a versatile mmWave beam prediction model. Early attempts to incorporate evolutionary algorithms (EAs) and reinforcement learning (RL) algorithms into NAS have focused on training probabilistic models to sample architectures with superior performance. Notably, previous works such as [26] and [32] have achieved remarkable results, requiring extensive computation resources for both searching and training. To expedite the search process, gradient-based NAS methods that utilize gradient descent optimization have been proposed [33], [34]. One prominent gradient-based NAS method is [35], which utilizes a continuous search space derived from the continuous relaxation of architecture representation.

In this work, inspired by the aforementioned neural architecture search (NAS) algorithms, we propose a novel method named as channel estimation neural network architecture search (CENAS). Our objective is to intelligently and automatically design a neural network architecture for channel estimation. By employing CENAS, even individuals lacking expertise in complex neural network knowledge are able to obtain highly efficient, customized, and intelligent channel estimation strategies. To the best of our knowledge, our work is the first application of NAS within the field of channel estimation. Notably, our work is primarily associated with prior works such as CDRN [16] and DARTS [35]:

- Our research pioneers the application of NAS in the domain of channel estimation, opening up novel avenues for the synergistic incorporation of neural networks into communication systems. In particular, we proposes a novel channel estimation NAS method called “CENAS,” which revolutionizes the design of neural network architectures in IRS-aided multi-user multiple-input single-output (MISO) systems. This approach obviates the necessity for laborious trial-and-error procedures, redefining the neural network design challenge as a continuous bilevel optimization problem, thus liberating it from conventional empirical reliance.
- To obtain the network finely tuned to the channel estimation task, we introduce a tailored search space designed explicitly for channel estimation. Coupled with a carefully chosen initialization strategy, this customization achieves an optimized equilibrium between network weights and architecture. These unique characteristics of CENAS guarantee its feasibility.

- The incorporation of the truncated back-propagation optimization algorithm plays a pivotal role in fostering both convergence and performance within the CENAS framework. This integration effectively mitigates the prevalent issue of the skip-collapse phenomenon [36] commonly encountered in many NAS algorithms based on DARTS [35]. By doing so, we not only improve the training stability but also ensure that CENAS consistently designs high-quality channel estimation neural networks.
- The CENAS framework facilitates the construction and optimization of neural networks that exhibit competitive performance across various wireless communication systems, accommodating variations in SNR, number of users, and antennas. To enhance the credibility and applicability of our research, we extend our investigations by incorporating additional simulations that leverage the more intricate and authentic DeepMIMO dataset [37]. Extensive numerical results validate the effectiveness of CENAS, demonstrating its superiority over traditional methods such as LS method and linear minimum mean squared error (LMMSE) method, as well as deep learning-based approaches, particularly in low SNR environments.

The rest of this paper is organized as follows. Section II provides the system model and problem formulation for channel estimation. Section III formulates the problem of neural network architecture search and presents our proposed CENAS method. In Section IV, theoretical analysis is provided and presents the results of numerical simulations. Finally, Section VII is the conclusion.

Notation: We utilize the following notations throughout the paper: $\mathbf{A}$ represents a matrix, $\mathbf{a}$ denotes a vector, and a represents a scalar. The Frobenius norm of a matrix is denoted as $\|\cdot\|_F$. $\mathcal{CN}(\mathbf{m}, \mathbf{R})$ represents a complex Gaussian distribution with mean $\mathbf{m}$ and covariance $\mathbf{R}$. $\nabla_\alpha \mathcal{L}$ denotes the gradient of function $\mathcal{L}$ with respect to the variable α.

II. System Model and Channel Estimation.

In this section, we consider an IRS aided multi-user MISO communication system similar to [16], and formulate the channel estimation problem. As shown in Fig. 1, the system includes an IRS with N passive reflecting elements that assists in the communication between an access point (AP) and K users. We assume that a time division duplex (TDD) protocol is adopted in the system. In this work, we formulate the proposed channel estimation problem as a deep learning-based denoising problem. Next, we provide a detailed explanation of the system model and the channel estimation problem.

A. System Model

In this system, an AP and K users are equipped with an antenna array consisting of M elements and a single antenna, respectively. To improve communication performance, an IRS consisting of N passive reflecting elements is deployed for the signal transmission. Due to the channel reciprocity property in TDD systems, the downlink channel state information could be acquired by estimating the uplink CSI. The baseband equivalent channels of the user k-IRS link, the IRS-AP link and the user k-AP link are denoted by $\mathbf{h}_{k,r} \in \mathbb{C}^{N\times 1}$, $\mathbf{G} \in \mathbb{C}^{M\times N}$ and $\mathbf{h}_{k,A} \in \mathbb{C}^{M\times 1}$, respectively. Note that the subscripts r and A represent IRS and AP, respectively. We define a diagonal matrix $\mathbf{\Theta} \triangleq \mathrm{diag}(\mathbf{r}) \in \mathbb{C}^{N\times N}$ as the reflection-coefficients matrix of IRS with

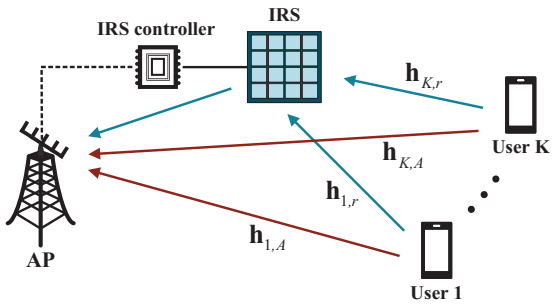


Fig. 1. The uplink of the considered system model.

$$\mathbf{r} = \left[\beta_1 e^{j\theta_1}, \beta_2 e^{j\theta_2}, \cdots, \beta_N e^{j\theta_N}\right], \tag{1}$$

where $0 \le \beta_n \le 1$ and $0 \le \theta_n < 2\pi$ are the amplitude and phase shift of the n-th element, respectively. The cascaded channel from the k-th user to AP via IRS is able to be expressed as $\mathbf{G}\mathbf{\Theta}\mathbf{h}_{k,r} \in \mathbb{C}^{M\times 1}$. By applying the change of variables, we have

$$\mathbf{G}\mathbf{\Theta}\mathbf{h}_{k,r} = \mathbf{G}\mathrm{diag}(\mathbf{h}_{k,r})\,\mathbf{r}. \tag{2}$$

Thus, the effective channel from the k-th user to AP can be represented as

$$\mathbf{h}_{k,A} + \mathbf{G}\mathrm{diag}(\mathbf{h}_{k,r})\,\mathbf{r} = \mathbf{U}_k\bar{\mathbf{r}}, \tag{3}$$

with

$$\mathbf{U}_k = \left[\mathbf{G}\mathrm{diag}(\mathbf{h}_{k,r}), \mathbf{h}_{k,A}\right], \forall k, \tag{4}$$

and $\bar{\mathbf{r}} = [\mathbf{r}; 1]$. Therefore, the channel estimation objective is to estimate the matrix $\mathbf{U}_k$.

In the following, we utilize an uplink pilot transmission scheme for channel estimation. During the channel estimation phase, IRS generates $P\,(\ge N+1)$ different reflection patterns via P different phase shift matrices, denoted by $\alpha = [\mathbf{\Theta}_1, \mathbf{\Theta}_2, \cdots, \mathbf{\Theta}_P]^T$. For each user k, a pilot sequence $\mathbf{u}_k = [u_{k,1}, u_{k,2}, \cdots, u_{k,L}]^T$ with a length L ($L \ge K$), is adopted for each IRS reflection pattern. Note that each two pilot sequences should be orthogonal for ease of distinguishing different users, i.e., $\mathbf{u}_m^H\mathbf{u}_n = 0, \forall m \ne n$, and $\mathbf{u}_m^H\mathbf{u}_m = P_t L$, where P_t is the power of each pilot symbol. Therefore, the received pilot signal vectors during the p-th sub-frame at AP are capable of being expressed as

$$\mathbf{S}_p = \sum_{k=1}^{K} \mathbf{U}_k\bar{\mathbf{r}}_p\mathbf{u}_k^H + \mathbf{V}_p, \tag{5}$$

with $\mathbf{S}_p = [\mathbf{s}_{p,1}, \mathbf{s}_{p,2}, \cdots, \mathbf{s}_{p,L}] \in \mathbb{C}^{M\times L}$, and $\mathbf{V}_p = [\mathbf{v}_{p,1}, \mathbf{v}_{p,2}, \cdots \mathbf{v}_{p,L}]$, where $\mathbf{s}_{p,l} \in \mathbb{C}^{M\times 1}$ is the l-th received pilot signal in the p-th sub-frame and $\mathbf{v}_{p,l} \in \mathbb{C}^{M\times 1}$ is the l-th noise vector at AP in the p-th sub-frame. Generally, we assume $\mathbf{v}_{p,l}$ to be a circularly symmetric complex Gaussian (CSCG) random vector with $\mathbf{v}_{p,l} \sim \mathcal{CN}\left(\mathbf{0}, \sigma_v^2\mathbf{I}_M\right)$, where σ_v^2 is the

noise variance of each antenna at AP. By using the orthogonal property of the pilot sequences of different users, we have

$$\mathbf{w}_{p,k} = \mathbf{U}_k \bar{\mathbf{r}}_p + \mathbf{z}_{p,k}, \tag{6}$$

where $\mathbf{w}_{p,k} = \frac{1}{P_t}\mathbf{S}_p \mathbf{u}_k$ denotes the received signal vector at AP form the k-th user in the p-th sub-frame, and $\mathbf{z}_{p,k} \sim \mathcal{CN}\big(\mathbf{0}, \frac{1}{P_t}\sigma_v^2 \mathbf{I}_M\big)$. After receiving P sub-frames at AP, we have

$$\mathbf{W}_k = \mathbf{U}_k \mathbf{R} + \mathbf{Z}_k, \tag{7}$$

where $\mathbf{W}_k = [\mathbf{w}_{1,k}, \mathbf{w}_{2,k}, \cdots, \mathbf{w}_{P,k}] \in \mathbb{C}^{M\times P}$ is the result of stacking $\mathbf{w}_{p,k}$ from all the P sub-frames, $\mathbf{R} = [\bar{\mathbf{r}}_1, \bar{\mathbf{r}}_2, \cdots, \bar{\mathbf{r}}_P] \in \mathbb{C}^{(N+1)\times P}$ and $\mathbf{Z}_k = [\mathbf{z}_{1,k}, \mathbf{z}_{2,k}, \cdots, \mathbf{z}_{P,k}] \in \mathbb{C}^{M\times P}$. Without loss of generality, $\mathbf{R}$ is designed as a discrete Fourier transform (DFT) form, i.e.,

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_P & \cdots & W_P^{P-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_P^N & \cdots & W_P^{N(P-1)} \end{bmatrix} \in \mathbb{C}^{(N+1)\times P}, \tag{8}$$

with $W_P = e^{j2\pi/P}$.

B. Channel Estimation

Channel estimation aims to recover $\mathbf{U}_k$ by exploiting $\mathbf{W}_k$ and $\mathbf{R}$. Numerous methods [15], [16], [22] have been proposed for solving this problem. Conventionally, LS method is a practical method if no prior knowledge of channel is available, i.e.,

$$\mathbf{U}_k^{LS} = \mathbf{W}_k \mathbf{R}^{\dagger}, \tag{9}$$

where $\mathbf{U}_k^{LS}$ denotes the estimated $\mathbf{U}_k$ by adopting the LS estimator and $\mathbf{R}^{\dagger} = \mathbf{R}^H\big(\mathbf{R}\mathbf{R}^H\big)^{-1}$ is the pseudoinverse of $\mathbf{R}$. LS method treats the channel as a deterministic but unknown constant.

In contrast to neural networks, LS method relies on prior knowledge of the channel statistics, such as the channel covariance matrix, which is often difficult to acquire in practical scenarios [14]. Furthermore, the performance of LS method deteriorates significantly in the presence of noises, and it is unable to capture the complex non-linear relationships between the received signal and the channel coefficients [16]. On the other hand, neural networks learn complex non-linear mappings from the input signal to the channel coefficients, without the need for prior knowledge of the channel statistics, and are capable of handling noise effectively by learning robust representations of the input data [14].

DL methods are focused on designing network architectures that are able to achieve high channel estimation accuracy. Several DL-based methods have been proposed in recent years [14], [16], [22]. Similar to [16], first we exploit the coarse estimated value by (9), then the channel estimation problem is transformed into a denoising problem, i.e., recovering $\mathbf{U}_k$ from $\mathbf{U}_k^{LS}$, where

$$\mathbf{U}_k^{LS} = \mathbf{U}_k + \mathbf{Z}_k \mathbf{R}^{\dagger}. \tag{10}$$

Following [16], We develop a DL-based channel estimation algorithm to solve the following problem

$$\hat{\mathbf{U}}_k = \mathcal{F}_{\alpha,\omega}(\mathbf{U}_k^{LS}), \tag{11}$$

where $\mathcal{F}_{\alpha,\omega}(\cdot)$ indicates the neural network, α stands for the neural network architecture and ω denotes the neural network weight parameters. According to MMSE criterion, the channel estimation problem can be formulated as

$$\min_{\mathcal{F}_{\alpha,\omega}} E\left[\left\|\mathbf{U}_k - \mathcal{F}_{\alpha,\omega}(\mathbf{U}_k^{LS})\right\|_F^2\right]. \tag{12}$$

From (12),clearly, the neural networks determine the performance of the aforementioned channel estimation problem. Therefore, the channel estimation task is transformed into the network architecture construction and training. For instance, in [16], a neural network named CDRN is introduced that takes advantage of the deep residual network architecture for feature extraction and denoising. The ResU-Net is presented in [14], where skip connections are used to fuse features of different scales. Additionally, in [22], a network architecture named multiple residual dense network(MRDN) is proposed, which is inspired by several recent image restoration networks that utilized residual dense network architecture to restore the channel matrix. However, previous methods aimed to find the best network weight parameters under a given architecture through training. Different from them, we are motivated by the concept of learning to learn and seek to simultaneously find the best architecture and parameters. To this end, we propose a network architecture search method called CENAS for the automatic design of high-performance neural networks for the channel estimation task.

III. CENAS FRAMEWORK

In this section, the channel estimation problem is formulated as a deep learning-based denoising problem. The higher performance and better task compatibility of the neural network designed by CENAS are attributed to the appropriate search space and efficient gradient-based search strategy. Consistent with previous work [35], our framework relies on the continuous relaxation of the network architecture representation. In the subsequent sections, we present the problem formulation, CENAS architecture, and the proposed CENAS algorithm. Moreover, we analyze the convergence of the CENAS framework, similar to the approach described in [38], thereby providing valuable insights into the convergence behavior of our proposed method.

A. Problem Formulation

The previous researches [14], [16], [22] utilize training to find the best network weight parameters under the manually designed network architecture. Motivated by the idea of learning to learn, we are looking for the best architecture and parameters simultaneously. On the other hand, the denoise neural networks often cascade the denoise module repeatedly to construct the neural network [16], [22]. To obtain the best denoise neural network module, we propose the CENSA framework to alleviate the burden of tedious and heuristic human design to discover the adaptive neural network for channel estimation.

We denote the architecture set as $\mathcal{A} = \{\alpha_1, \alpha_2, \cdots, \alpha_M\}$, where α_m represents one specific network architecture type.

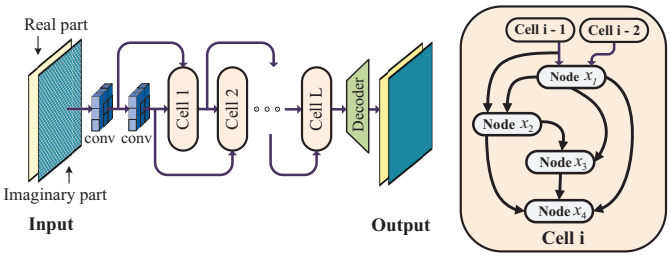


Fig. 2. Left: CENAS architecture. Right: Directed acyclic graph (DAG) representation of the cell.

Naturally, the design and training of the neural network in equation (12) can be modeled as a bilevel optimization problem:

$$\min_{\alpha} \mathcal{L}_{val} = E\left[\left\|\mathbf{U}_k - \mathcal{F}_{\alpha,\omega^*(\alpha)}\right\|_F^2\right], \tag{13a}$$

$$\text{s.t. } \omega^*(\alpha) = \text{argmin}_{\omega}\, \mathcal{L}_{train}, \tag{13b}$$

where $\omega^*(\alpha)$ is the optimal weight of the network architecture α optimized by training to achieve the minimum training loss (the loss used in our work is presented in (25)). The architecture α is optimized based on the optimized weight ω^*. By solving the above optimization problem, prior knowledge that helps the estimation of channel $\mathbf{U}_k$ from $\mathbf{U}_k^{LS}$ can be obtained.

B. CENAS Search Space

As illustrated in Fig. 2, the neural networks constructed by CENAS is a chain-structured neural network consisting of three modules: Two convolution layers, different denoise cells $\{C_1, C_2, \cdots, C_{L-1}\}$ and a decoder module. Two 3×3 convolution layers are cascaded to generate two input feature maps for the first cell. The last decoder module gradually recovers channel details and spatial dimension to the channel's original dimensions consisting of a sequence of operations: a single layer of full connection, two layers of 3×3 separable convolutions, and a final 3×3 convolution.

Obviously, the different denoise cell types and sequences offers a wide array of network architectures for algorithm to explore. This enables the discovery of neural networks that are specifically designed to enhance channel estimation performance. Specifically, each cell can be represented by a directed acyclic graph (DAG) in Fig. 2 right which has one single output and takes previous two cells outputs as its inputs. In one cell, the nodes $\{x_1, x_2, \cdots, x_{N-1}\}$ stand for feature maps, and each edge $o^{(i,j)}(x) \in \mathcal{O}$ is an operation that transforms the node x_i to x_j. The output of the cell C_l $(0 \le l \le L)$ is a concatenation of all intermediate nodes and it can be expressed as $C_l = \sum_{x_i \in C_l} Concat(x_i)$. Each intermediate node gets feature maps from all its predecessors: $x_j = \sum_{i<j} o^{(i,j)}(x_i)$.

In contrast with the immovable architecture in [14], [16], [22], our search space is flexible and renewable that incorporates several updated hand-crafted architectures, e.g., skip connections and separable convolutions. These two operations are used to build deeper networks and speed up the convergence. In this paper, each convolution operation is a combination of convolution, batch normalization, and leaky rectified linear unit. The operations set $\mathcal{O}$ contains 5 types of operation:

- skip connection;
- zero;
- 3×3 convolution;
- 3×3 separable convolution;
- 3×3 dilation convolution;

Different from all aforementioned methods [26], [32], [35], we avoid to use the pooling and downsampling operations in set $\mathcal{O}$ but select dilation convolutions instead [39]. The pooling and downsampling operations is not the most suitable approach for denoising tasks, primarily due to the loss of crucial spatial details, textures, and small-scale structures [40]. Achieving a delicate balance between noise reduction and information preservation remains paramount for optimal results in denoising. Consequently, the dilated convolutions retain adequate information from channel matrix feature maps and adjust the receptive field of the channel matrix feature maps adaptively during the architecture search process [41]. The zero operation not only determines the types of feature map transformations, but also controls the edge connections in the cell's DAG. With the definition of search space illustrated in Fig. 3, architecture optimization is transformed into optimizing the operation selection $o^{(i,j)}(x_i)$ on each edge x_i in the cells l, i.e.,

$$\min_{o^{(i,j)}} \mathcal{L}_{val}(\omega^*(o^{(i,j)}), o^{(i,j)}), \tag{14a}$$

$$\text{s.t. } \omega^*(o^{(i,j)}) = \text{argmin}_{\omega}\, \mathcal{L}_{train}. \tag{14b}$$

C. Continuous Relaxation

Initially, the difficulty of the optimization problem in (14) is determined by the search space. This problem remains non-continuous due to the discrete set $\mathcal{O}$. And the diversity of cells increases the complexity of the search space. With the assumption that the cell number and node number are L and N, there are $5^{LN(N-1)/2}$ networks in the search space. If we choose $L = 3, N = 3$, resulting in 1.9×10^6 possible networks. It is tricky to find an algorithm to efficiently exploit this enormous discrete search space. Apart from this, optimization of the (14a) requires training each architecture $\alpha \in \mathcal{A}$ to converge, which is computationally expensive and time consuming.

Following [35], we adopt continuous relaxation to the problem mentioned before. This approach relaxes the categorical choice of a specific operation to a soft one. As shown in Fig. 3, we use the *softmax* function over operations on each edge to make the search space continuous. Discrete categorical operations in the cell are relaxed to a mixed operation as follow

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp\left(\alpha_o^{(i,j)}\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\alpha_{o'}^{(i,j)}\right)} o(x). \tag{15}$$

From node x_i to node x_j, the mixed operation weights are parameterized by $\alpha^{(i,j)}$ with the dimension of $|\mathcal{O}|$. Each operation $\bar{o}^{(i,j)}$ associated with a continuous coefficient $\alpha_o^{(i,j)}$. The output of each layer is obtained by computing the expectation

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2024.3374352

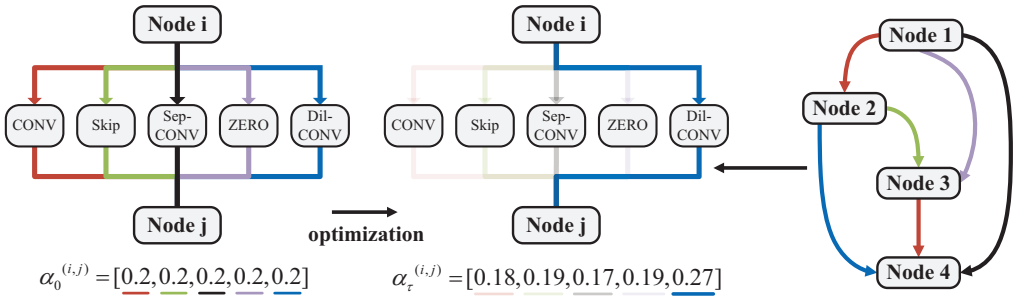


Fig. 3. Illustration of an edge optimization. Different color lines represent various operations in search space. After optimization, the operations with small α weight are abandoned represented by transparent lines.

over all possible outputs of the candidate operations. It is obvious that the task of architecture search transformed into learning a set of continuous variables $\alpha^{(i,j)}$ and can also be regarded as learning a distribution of operations for the channel estimation task.

Different from applying evolution or reinforcement learning to a discrete and non-differentiable search space, the problem is transformed into a continuous bilevel optimization problem after relaxation, which allows the architecture search to be performed in a differentiable manner [36]:

$$\min_{\alpha} \mathcal{L}_{val}(\omega^*(\alpha), \alpha), \tag{16a}$$

$$\text{s.t. } \omega^*(\alpha) = \text{argmin}_{\omega} \mathcal{L}_{train}. \tag{16b}$$

The variables ω and α correspond to the network weight and architecture, respectively. The upper-level objective is denoted by $\mathcal{L}_{val}$ while the lower-level objective is represented by $\mathcal{L}_{train}$, where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ are the loss function on training dataset and validation dataset.

D. Truncated back-propagation Optimization

It should be noted that (16) poses a significant challenge due to the intricate interdependence between the upper-level problem and ω. This challenge is further compounded when both ω and α are high-dimensional, thereby rendering the conventional black-box optimization methods ineffectively, such as grid/random search and Bayesian optimization [38].

In the current learning and vision literature, a popular assumption is that the lower-level objective (16b) is supposed to satisfy the lower-level singleton (LLS) condition [42]. Specifically, if ω^* is a singleton for all α belonging to the feasible set $\mathcal{A}$. The gradient of $\mathcal{L}_{val}$ with respect to the upper-level variable α can be expressed as follows

$$\frac{\partial \mathcal{L}_{val}(\omega, \alpha)}{\partial \alpha} = \frac{\partial \mathcal{L}_{val}\left(\omega^*(\alpha), \alpha\right)}{\partial \alpha} + \underbrace{\left(\frac{\partial \omega^*(\alpha)}{\partial \alpha}\right)'}_{\text{Jacobian}} \frac{\partial \mathcal{L}_{val}\left(\omega^*(\alpha), \alpha\right)}{\partial \omega(\alpha)}. \tag{17}$$

Similar with [43], we regard the iterative optimization algorithm utilized to solve the lower-level problem as a dynamical system. The updating rule of $\omega(\alpha)$ is defined as

$$\omega_t = \mathbf{\Psi}_t(\omega_{t-1}, \alpha). \tag{18}$$

According to the chain rule, the formulation of Jacobian can be expressed as

$$\frac{\partial \omega_t}{\partial \alpha} = \frac{\partial \Psi_t\left(\omega_{t-1}, \alpha\right)}{\partial \omega_{t-1}} \frac{\partial \omega_{t-1}}{\partial \alpha} + \frac{\partial \Psi_t\left(\omega_{t-1}, \alpha\right)}{\partial \alpha}. \tag{19}$$

Then, let

$$\mathbf{A}_t = \frac{\partial \Psi_t\left(\omega_{t-1}, \alpha\right)}{\partial \omega_{t-1}}, \mathbf{B}_t = \frac{\partial \Psi_t\left(\omega_{t-1}, \alpha\right)}{\partial \alpha}, \tag{20}$$

plugging (19) and (20) into the equation (17), we have

$$\frac{\partial \mathcal{L}_{val}(\omega, \alpha)}{\partial \alpha} = \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \alpha} + \underbrace{\sum_{t=0}^{T} B_t A_{t+1} \cdots A_T}_{\text{Jacobian}} \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega}, \tag{21}$$

where $t \geq 0$, and $B_0 = \frac{\partial \Psi_0}{\partial \alpha}$. The computation of Jacobian $\nabla_\alpha \omega^*(\alpha)$ can be implemented by backward propagation [44]. We define

$$\mathbf{g}_t = \frac{\partial \mathcal{L}_{val}(\omega_t^*, \alpha)}{\partial \alpha}, \lambda_t = \frac{\partial \mathcal{L}_{val}\left(\omega_t^*, \alpha\right)}{\partial \omega_t}, \tag{22a}$$

$$\mathbf{g}_{t-1} = \mathbf{g}_t + \mathbf{B}_t' \lambda_t, \lambda_{t-1} = \mathbf{A}_t' \lambda_t, \tag{22b}$$

with $t = T, \cdots, 0$. Finally we can get $\mathbf{g}_{-1} = \frac{\partial \mathcal{L}_{val}(\omega^*(\alpha), \alpha)}{\partial \alpha}$.

To mitigate the time and space complexities associated with computing approximate gradients $\nabla_\alpha \mathcal{L}_{\text{val}}(\omega, \alpha)$, we opts for few-step reverse-mode automatic differentiation (RAD) while disregarding long-term dependencies, which is referred to as truncated RAD [38]. In particular, we employ a truncated back-propagation algorithm and utilize the intermediate variable $\mathbf{g}_{T-M}$ to construct an approximation of the gradient, i.e.,

$$\frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} \approx \mathbf{g}_{T-M} = \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \alpha} + \sum_{t=T-M+1}^{T} B_t A_{t+1} \cdots A_T \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega}. \tag{23}$$

Therefore, the prohibitive optimization in (21) evaluating the architecture gradient from scratch can be divided approximately into two steps: First, the neural network weight ω is optimized with the truncated RAD based on the current architecture α_τ. Second, the architecture weight α_τ is updated

Algorithm 1 Truncated RAD

Input: The upper-level variable at the current stage α, the lower-level initialization ω_0 and the truncated steps M.
Output: The approximate gradient of $\mathcal{L}_{val}$ with respect to α, i.e., $\frac{\partial \mathcal{L}_{val}}{\partial \alpha}$ and the ω_T.

$\omega_0 = \Psi_0(\alpha)$.
for $t = 1, \cdots, T$ **do**
  $\omega_t = \Psi_t(\omega_{t-1}, \alpha)$.
end for
$\mathbf{g}_T = \frac{\partial F(\alpha, \omega_T)}{\partial \alpha}$ and $\lambda_T = \frac{\partial F(\alpha, \omega_T)}{\partial \omega_T}$.
for $t = T, \cdots, T - M$ **do**
  $\mathbf{g}_{t-1} = \mathbf{g}_t + \mathbf{B}_t' \boldsymbol{\lambda}_t$ and $\boldsymbol{\lambda}_{t-1} = \mathbf{A}_t' \boldsymbol{\lambda}_t$.
end for
return $\frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} \approx \mathbf{g}_{T-M}$ and the ω_T.

based on the optimized weight ω with the loss function $\mathcal{L}_{train}$, i.e.,

$$\Psi_t = \omega_t - \gamma \nabla_\omega \mathcal{L}_{train}(\omega_t, \alpha_\tau), \tag{24a}$$

$$\alpha_{\tau+1} = \alpha_\tau - \eta \nabla_\alpha \mathcal{L}_{val}(\omega^*(\alpha_\tau), \alpha_\tau), \tag{24b}$$

where $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ are the loss function on training dataset and validation dataset. η is the learning rates of the architecture parameters. γ is the inner optimization learning rate.

E. CENAS Algorithm

Overall, the CENAS algorithm illustrated in Algorithm 2 consists of the pre-training phase and architecture search phase, where the details are given as follows.

1) Dataset Preparation: We perform multiple Monte Carlo simulations to generate the dataset. Dataset $\mathcal{D} = \{(\mathbf{U}_i, \tilde{\mathbf{U}}_i)\}_{i=1}^{N_t}$, $\mathbf{U}_i = [\mathrm{Re}\{\mathbf{U}_i\}, \mathrm{Im}\{\mathbf{U}_i\}] \in \mathbf{R}^{M \times (N+1) \times 2}$ denote the labels, i.e. the real and imaginary parts of the channel include 60,000 samples. $\tilde{\mathbf{U}}_i$ denote the input channel matrix from the noisy pilot-based observations $\mathbf{U}_k^{LS}$. Similar to [16], we divide the real part and imaginary part of the channel matrix into two dimensions. After pre-training phase, the dataset $\mathcal{D}$ is divided into two same size components of $\mathcal{D}_\omega$ and $\mathcal{D}_\alpha$ randomly for optimizing α and ω, respectively.

2) Optimization Balance with Good Initialization: In several initial attempts to apply the CENAS method to this task, the algorithm often show non-convergence and poor performance. Without pre-training, neural networks tend to exhibit a strong preference for skip connections, primarily due to their initial dominant position, which extends the convergence time of the search and traps the architecture α in a local optimum [45]. Moreover, allowing for the quantities of ω and the α in the search space are substantially different, the additional optimizations iterations of the weight parameters ω are necessary. Without such iterations, these same optimization iterations can contribute to overfitting of α and non convergence of ω. Through careful analysis of the experimental data, we arrive at the following conclusions: proper initialization of the network weight and optimization balance are critical to the architecture search. The experiment are illustrated in numerical simulations.

Algorithm 2 CENAS

Input: Epoch = 0, Inner optimization steps:$t = 0, \cdots, T-1$, Data $\mathcal{D} = \{(\mathbf{U}_i, \tilde{\mathbf{U}}_i)\}_{i=1}^{N_t}$, initialized network weight ω_0, initialized architecture weight α_0, and the learning rates η, β, γ.
Output: Network architecture weight α^*.

1: **Pre-trianing Phase:**
2: Construct pretrain network based on ω_0.
3: **while** Epoch $\leq$ 10 **do**
4: Update ω using $\omega_{t+1} = \omega_t - \beta \nabla_\omega \mathcal{L}_{train}(\omega_t, \alpha)$.
5: **end while**
6: Obtain the pretrained network weight $\omega_{pretrain}$.
7: **Architecture Searching Phase:**
8: Generate corresponding datasets $\mathcal{D}_\omega$ and $\mathcal{D}_\alpha$.
9: Construct network based on the $\omega_{pretrain}$.
10: **while** α not converged **do**
11: Calculate the $\nabla_\alpha \mathcal{L}_{val}(\omega^*, \alpha_\tau)$ with truncated RAD algorithm and update ω_t with (24a) using dateset $\mathcal{D}_\omega$.
12: Update $\alpha_{\tau+1}$ using (24b) with the dateset $\mathcal{D}_\alpha$.
13: Let the optimal weights ω_{T-1} of the α_τ as the initialized weights ω_0 of subsequent iterations: $\omega_0(\alpha_{\tau+1}) = \omega_{T-1}(\alpha_\tau)$.
14: **end while**
15: Derive the final network based on α^*.

As depicted in Algorithm 2, we initiate the network weight ω_0 and allocate 10 epochs for pre-training the supernet. During this pre-training phase, each operation candidate's weights ω in the supernet are pre-trained using the loss function $\mathcal{L}_\omega$, which is given by

$$\mathcal{L}_\omega = \frac{1}{2N_t} \sum_{i=1}^{N_t} \left\| \tilde{\mathbf{U}}_i - \mathbf{U}_i \right\|_F^2, \tag{25}$$

where N_t is the number of training samples. After pre-training phase, all operations can compete on a more equitable basis during the optimization process. This enables the networks to achieve higher levels of accuracy and ensures a relatively fair starting point during the architecture search phase.

3) Architecture Optimization and Construction: In Algorithm 2, the process of architecture search is carried out through alternative optimization, which involves the use of (24) and following steps 8-14. After the pretraining phase, the network architecture is constructed using the initialized network architecture weight α_0, which is based on the pretrained network weight $\omega_{pretrain}$. In order to uphold the asymptotic convergence of our proposed methodology, we utilize the truncated RAD algorithm in step 11 to compute the gradient of the validation loss, $\nabla_\alpha \mathcal{L}_{val}(\omega, \alpha)$, using the dataset $\mathcal{D}_\omega$. Subsequently, a gradient descent is performed by updating α with the computed gradient $\nabla_\alpha \mathcal{L}_{val}(\omega^*, \alpha_\tau)$ using the dataset $\mathcal{D}_\alpha$. Finally, taking inspiration from [35], [46], we adopt a strategy to ensure the continuity of $\omega(\alpha)$ in step 13. This step allows us to inherit the optimal weights from the last α_τ optimization, denoted as $\omega_{t=T}(\alpha_\tau)$, and use them as the initialized weights $\omega_{t=0}(\alpha_{\tau+1})$ for subsequent iterations of $\alpha_{\tau+1}$. This approach significantly reduces the training time

for different networks, as it eliminates the need to train a new network from scratch.

Our approach calcualtes $\mathcal{L}_\alpha$ in a manner identical to (25) during the architecture searching phase. This is depicted in the second directed acyclic graph (DAG) in Fig. 3. We subsequently derive a network based on the optimal architecture α^* that selects operations based on the mixed operation outlined in (15). The optimized network architecture is then trained for a period of 60 epochs.

The paper [35] proposes an approach to approximate the gradient of $\nabla_\alpha \mathcal{L}_{val}(\omega^*(\alpha), \alpha)$ as

$$\nabla_\alpha \mathcal{L}_{val}(\omega^*(\alpha), \alpha) \approx \nabla_\alpha \mathcal{L}_{val}(\omega - \gamma \nabla_\omega \mathcal{L}_{train}(\omega, \alpha), \alpha). \tag{26}$$

Here, $\omega^*(\alpha)$ is approximated by adapting ω using only a single training step, rather than solving the inner optimization problem completely by training until convergence. Nonetheless, the utilization of the aforementioned approach results in an error between ω^* and $\omega - \gamma \nabla_\omega \mathcal{L}_{train}(\omega, \alpha)$, leading to a drastic increase in the number of skip connections during the architecture search phase. This, in turn, yields unsatisfactory performance after a certain number of search epochs, as described in [47] as the phenomenon of "performance collapse." Different from [35], our method utilizes the truncated RAD algorithm to calculate the gradient $\nabla_\alpha \mathcal{L}_{val}(\omega^*(\alpha), \alpha)$ during the iterative optimization of α and ω. The number of skip connections in DARTS increases significantly after a certain number of search epochs, which is in contrast to the typical fluctuations observed in the truncated RAD algorithm.

The implementation of the truncated RAD algorithm plays a significant role in promoting convergence and feasibility in CENAS framework. Moreover, our improvement can be attributed to the adoptions of an appropriate search space and an efficient gradient-based search strategy, which enable the CENAS framework to generate high-performance neural networks with superior task adaptability. The computational complexity of the neural network generated by CENAS during the testing phase is expressed as

$$\mathcal{O}(3^2 M(N+1)(\sum_{l=0}^{L-1} D_l + D_{conv1,2} + D_{decoder})), \tag{27}$$

where D_l, $D_{conv1,2}$, $D_{decoder}$ denote the numbers of convolution operations for the cell C_l, two convolution layers at the beginning, decoder module before the output, respectively. Note that three kinds of convolution operations in the set $\mathcal{O}$ have the same kernel size 3.

F. Theoretical Analysis of Gradient Optimization in CENAS.

The objective of CENAS is to identify the architecture parameters α for the optimization problem in (16), resulting in an approximate solution $\omega^*(\alpha)$ with low cost $\mathcal{L}_{val}$. This framework is a specific instance of the bilevel optimization problem, in which the upper-level objective $\mathcal{L}_{val}$ is not directly dependent on α [38]. This section analyzes the theoretical properties of CENAS. Follwing [38], [44], we prove that under local strong convexity assumptions, our methods can converge on average to an σ-stationary point. The proof is given in Appendix A.

Lemma 1. *In the considered IRS channel estimation problem, following [38], [44], we assume $\mathcal{L}_{train}$ is β-smooth, twice differentiable, and locally α-strongly convex in ω around $\{\omega_{T-M-1}, \ldots, \omega_T\}$. Let $\Psi_{t+1}(\omega_t, \alpha) = \omega_t - \gamma \nabla_w \mathcal{L}_{train}(\omega_t, \alpha)$. For $\gamma \leq \frac{1}{\beta}$, it holds*

$$\left\| \frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} - \mathbf{g}_{T-M} \right\| \leq 2^{T-M+1}(1-\gamma\alpha)^M \cdot \left\| \frac{\partial \mathcal{L}_{val}(\omega^*, \alpha)}{\partial \omega} \right\| \cdot M_B, \tag{28}$$

where $M_B = \max_{t \in \{0,\ldots,T-M\}} \|B_t\|$. In particular, if $\mathcal{L}_{train}$ is globally α-strongly convex, then

$$\left\| \frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} - \mathbf{g}_{T-M} \right\| \leq \frac{(1-\gamma\alpha)^M}{\gamma\alpha} \cdot \left\| \frac{\partial \mathcal{L}_{val}(\omega^*, \alpha)}{\partial \omega} \right\| \cdot M_B. \tag{29}$$

Following [38], the application of Lemma 1 yields the deduction that the optimization of α with $\mathbf{g}_{T-M}$ results in convergence to an approximate stationary point. The upper-level gradient in the τth iteration is denoted as $\nabla \mathcal{L}_{val}(\alpha_\tau)$.

Lemma 2. *We assume $\mathcal{L}_{train}$ be a smooth function [38], [42], [44] that is bounded below, and let $\epsilon < \infty$ be such that $\left\| \frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} - \mathbf{g}_{T-M} \right\| \leq \epsilon$. By employing $\mathbf{g}_{T-M}$ as a stochastic first-order approximation in the gradient descent update for α with a step size $\eta_\tau = O(1/\sqrt{\tau})$. After R iterations, we have the following convergence result:*

$$\mathbb{E}\left[\sum_{\tau=1}^{R} \frac{\eta_\tau \left\| \nabla \mathcal{L}_{val}(\alpha_\tau) \right\|^2}{\sum_{\tau=1}^{R} \eta_\tau} \right] \leq \widetilde{O}\left(\epsilon + \frac{\epsilon^2 + 1}{\sqrt{R}} \right). \tag{30}$$

This implies that, under the assumptions stated in Lemma1, learning with $\mathbf{g}_{T-M}$ converges to an ϵ-approximate stationary point, where $\epsilon = O\left((1-\gamma\alpha)^{-M}\right)$.

Furthermore, as the value of M increases, the bias becomes smaller. Hence, performing truncated back-propagation with $M = O(\log 1/\epsilon)$ is sufficient to update α effectively. If we set the values of $M = 1$ and $T = 1$, the CENAS method becomes equivalent to DARTS. We strike a balance between time and convergence by selecting appropriate value for M, which leads to sufficiently satisfactory results.

IV. NUMERICAL RESULTS

In this section, numerical simulations are carried out to evaluate the effectiveness of the proposed CENAS method. Firstly, convergence analysis is performed by monitoring the objective function value throughout the optimization process. A comparison is then made between the DARTS and our method by analyzing the architectures in terms of the number of skip-connections during the optimization process. Subsequently, we present the performance of our method under different signal-to-noise ratios (SNRs) and demonstrate the spectral efficiency using various channel estimation methods. Furthermore, simulations are conducted to assess the robustness of CENAS, taking variations in channel dimensionality

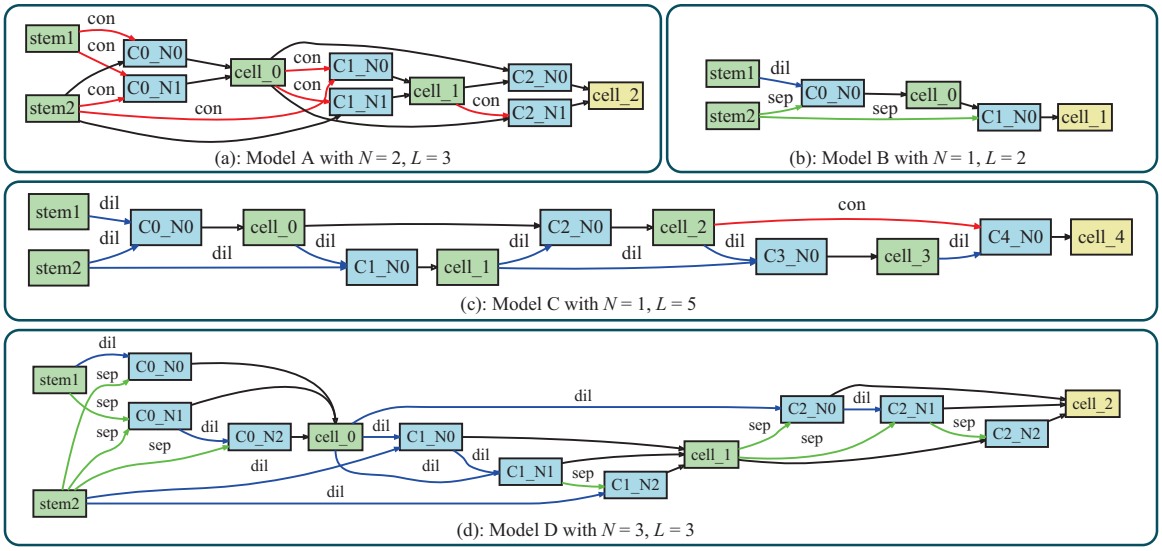


Fig. 4. The DAGs of four models. Dil, con, sep and distinct edge colors indicate the dilation convolution, convolution and separable convolution operations in the set $\mathcal{O}$. The stem1 and stem2 are the outputs of the CONV1 and CONV2 in Fig. 2. Different edge colors denote different operations.

and pilot configurations into account. Finally, we enhance the feasibility of our research by conducting simulation using a more realistic dataset [37].

A. Simulation Setup

We consider a IRS-aided multi-user MISO communication system with 6 UEs, where the AP is equipped with $M = 16$ antennas and the IRS consisting of $N = 64$ reflecting elements.

Similar to [16], the distances from the AP to the IRS, from the IRS to users, and from the users to AP are set as $d_{A,r} = 90$ m, $d_{r,k} = 16$ m, $\forall k$, $d_{k,A} = 100$ m, $\forall k$, respectively. The Rician channel model in this paper is assumed for all the channel links involved. In this case, the channel of IRS-AP link is formulated as

$$\mathbf{G} = \sqrt{\frac{\beta^{r,A}}{\beta^{r,A}+1}}\bar{\mathbf{G}} + \sqrt{\frac{1}{\beta^{r,A}+1}}\tilde{\mathbf{G}}, \tag{31}$$

where $\beta^{r,A}$ is the Rician factor of the IRS-AP channel, $\bar{\mathbf{G}}$ and $\tilde{\mathbf{G}}$ are the line-of-sight (LOS) and non-LOS (NLOS) components, respectively. The channels of other links can be generated via the same logic as defined in (31). Moreover, the path loss can be modeled as $L(d) = L_0(d/d_0)^{-\alpha}$, where $L_0 = -15$ dB is the path loss at the reference distance $d_0 = 10$ m and α is the path loss exponent. The path loss exponents of the user k-AP link, IRS-AP link and user k-IRS link are set as $\alpha^{k,A} = 3.6$, $\forall k$, $\alpha^{r,A} = 2.3$ and $\alpha^{k,r} = 2$, $\forall k$, respectively. In addition, the Rician factors are set as $\beta^{k,A} = 0$, $\forall k$, $\beta^{r,A} = 10$ and $\beta^{k,r} = 0$, $\forall k$, respectively. Note that the SNR in the simulation results is set as the transmit SNR, which is defined as $\mathrm{SNR} = P_t/\sigma_v^2$.

Multiple Monte Carlo simulations is conducted to generate the dataset. Each instance of the cascaded channel denoted as $\mathbf{U}$ has a shape of 65×16. A total of 60,000 instances are utilized as training samples, while 20,000 instances are reserved for testing. During the pretraining phase of the CENAS framework, we set the batch size as 64, and employ the stochastic gradient descent (SGD) optimizer to update the network weight parameter ω. To prevent the overfitting and promote regularization, a weight decay of 0.0003 is applied during the parameter updates. To archive more effective optimization and convergence during the pretraining process, cosine annealing strategy [48] is used to provide a smooth and gradual decrease in the learning rate from 0.005 to 0.0001.

After completing 10 epochs pretraining, the weight parameter ω_{pretrain} is obtained. In the architecture searching phase, we optimize ω_{pretrain} and the initialized α for 60 epochs, following steps 7-14 as outlined in Algorithm 2. During this phase, we set the values of M and T as 2 and 2, respectively, in the truncated RAD method. Additionally, we utilize a batch size of 64 and employ Adam optimizer to update the network architecture parameter α. Specifically, the learning rate for α update is set as 0.001, with the weight decay is set to 0.001. The hyperparameters for updating the network weight parameter ω remain consistent with those employed during the pretraining phase. Subsequently, a network is derived based on the optimal architecture α^* that selects operations based on the mixed operation outlined in (15).

TABLE I
DIFFERENT PERFORMANCE OF NETWORKS GENERATED BY VARIOUS HYPERPARAMETERS UNDER SNR = 0

Hyperparameters	$L = 2, N = 1$	$L = 5, N = 1$	$L = 3, N = 2$	$L = 3, N = 3$
NMSE(dB)	-9.39	-9.66	-9.91	-10.04
Total parameters	0.88×10^5	1.0×10^5	0.96×10^5	2.4×10^5

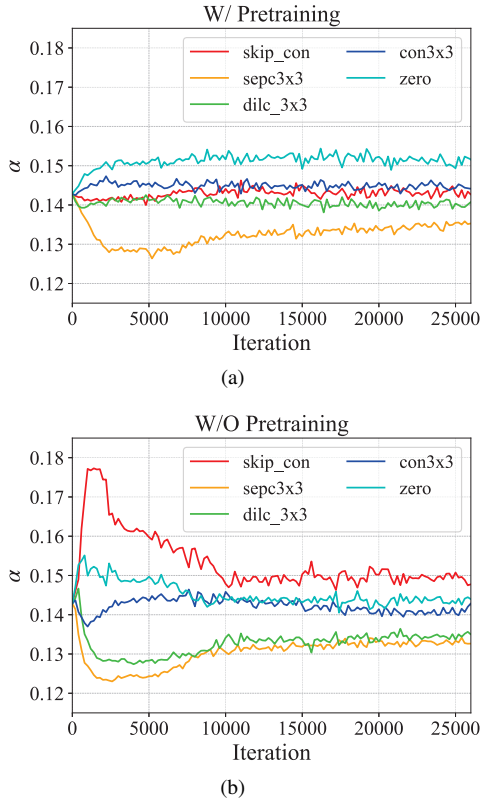


Fig. 5. Two different training schedules contribute to the different operation selections. One operation α value under two different training schedules. (a): After pre-training the ω, all operations compete more fairly during the optimization. (b): Without pre-training, skip connections become rapidly dominant in the beginning.

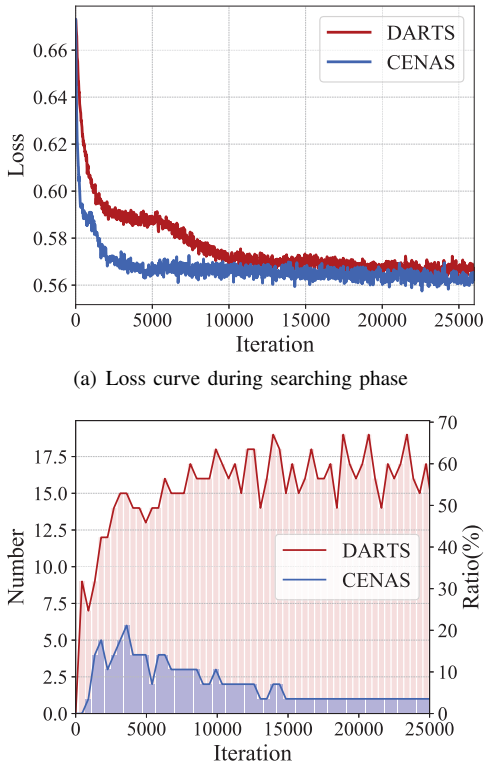


Fig. 6. (a): The $\mathcal{L}_{val}$ values for two different search approaches are compared, both using the same number of searching epoch. (b): The total number of skip-connections in the neural network during the search phase.

Finally, utilizing the network architecture discovered through the search process, as shown in the Fig. 4, we have obtained the optimized network architectures model D characterized by a cell number of $L = 3$ and a node number of $N = 3$. In our approach, we adopt a learning rate of 0.025 using the polynomial strategy [49], which dynamically adapts the learning rate during the training process. Moreover, we set the batch size to 64 and incorporate a weight decay of 0.0004. The 70 training epochs provide sufficient iterations for the network to converge and achieve a desirable performance that surpasses the other conventional and deep learning-based methods. All simulations are performed on Xeon(R) Gold 5118 CPU @ 2.30GHz and Tesla-V100 GPU.

In multiple experiments, the majority of search processes typically consumed slightly less time to converge compared to the time required for a full network training iteration. If the training time is denoted as T_{train}, then the totally CENAS time $T_{CENAS} \leq 2T_{train}$. This observation highlights that by selecting an appropriate search space, the time consumption for the CENAS method is approximately equivalent to training the network twice. It is evident that achieving optimal result with just two attempts is highly unlikely in the context of manual architecture design. Moreover, manually crafted architectures often suffer from a shortage of comprehensive comparative data, which hinders the effective assessment of their optimality.

Additionally, as illustrated in Fig. 4, four different network architectures are presented, each constructed by varying numbers of convolution operations D_l, and nodes connections. Model A with cell number $L = 3$, node number $N = 2$, model B with $L = 2$, $N = 1$, model C with $L = 5$, $N = 1$ and Model D is the selected network for the subsequent experiments with $L = 3$, $N = 3$. Regardless of the performance disparities among the four models, model B exhibits advantages in terms of the reduced convolution operations and the lower parameter count, rendering it more convenient for channel estimation task. On the other hand, model D, despite being computationally intensive, demonstrates the highest accuracy in estimating CSI. As shown in Table. I, the method generates high performance models with different parameters owing to the adjustable hyperparameters N, L.

B. Convergence and Skip collopse

In Fig. 5(a), the convolution operation pursues the skip connection hardly because of the initial dominant position, which extends the convergence time of the search and traps the architecture α in a local optimum [45]. In Fig. 5(b), all operations compete more fairly during the optimization after pre-training the ω. The networks adopting the zeros operation in Fig. 5(a) can achieve higher accuracy than the skip-connection operation in Fig. 5(b). Obviously, the experimental data shows that: proper initialization of the network weight and optimization balance are critical to the architecture search. The

bad optimization balance is accountable for the overfitting of α and premature convergence of ω.

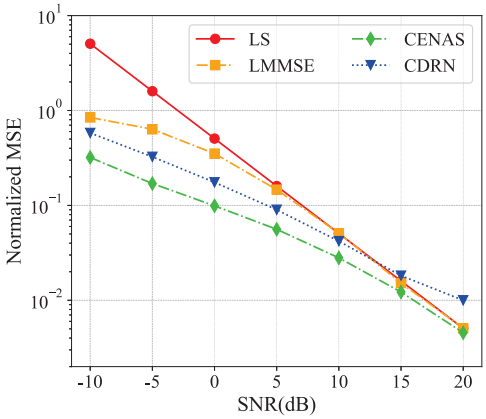
In the previous section, a comprehensive convergence theory analysis is conducted to evaluate the convergence behavior of the proposed method. To further assess the convergence characteristics, a numerical simulation is performed wherein we closely monitor the loss curves throughout the optimization process. The obtained $\mathcal{L}_{val}$ is depicted in Fig. 6(a), revealing the distinct convergence trends observed during the architecture searching phase. More precisely, when comparing the convergence behavior of our proposed method with DARTS method [35], we observe that our method exhibits a more rapid convergence in the initial stages of the optimization process. Furthermore, throughout the searching phase, our proposed method consistently converges towards the lower loss values. The convergence analysis highlights the superiority of our approach in terms of convergence stability and achieving more desirable optimization outcomes. Considering the loss in experiments, we strike a balance between time and convergence by selecting $M = 2$ and $T = 2$, which leads to sufficiently satisfactory results.

In DARTS method [35], we observe a phenomenon known as "skip collapse" [36] during the optimization process. As Fig. 6(b) shows, towards the end of the searching phase, the number of skip connections in the DARTS architecture experiences rapid fluctuations and accounts for nearly 60% of all operations. This observation indicates that DARTS becomes trapped in a local optima [36] and fails to achieve further convergence. However, despite the high frequency of skip connections, the performance of the DARTS architecture is found to be suboptimal. By employing the truncated RAD algorithm, we are able to mitigate the problem of convergence to local optima [36] associated with α optimization and attain more reliable and accurate outcomes. We can achieve better overall performance without encountering the issue like skip collapse.

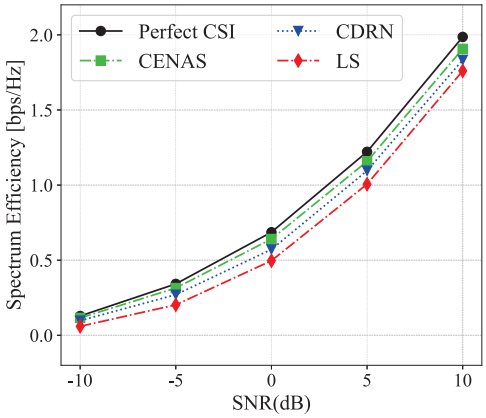
The limitations and potential risks associated with NAS present intriguing avenues for future research. Addressing the challenge of unintended, suboptimal architecture discovery involves developing more robust optimization objectives and defining stricter search space constraints, which can enhance NAS's reliability and applicability in communication scenarios [50]. Additionally, exploring countermeasures against potential adversarial interactions with NAS [51], including the development of safeguards and verification mechanisms, is essential to ensure the integrity, fairness, and security.

C. Performance with Different SNR and Spectral Efficiency

Fig. 7(a) presents a comprehensive comparative analysis of the normalized mean square error (NMSE) performance among various methods, including the proposed CENAS framework, the conventional LS method, the LMMSE method, and the DL-based method CDRN [16]. In most SNR ranges, the results demonstrate that the network architecture obtained by CENAS achieves an improvement in NMSE compared to CDRN, which also designs a neural network for the task. The crossover point around SNR = 15dB, between CDRN and



(a) NMSE performance.

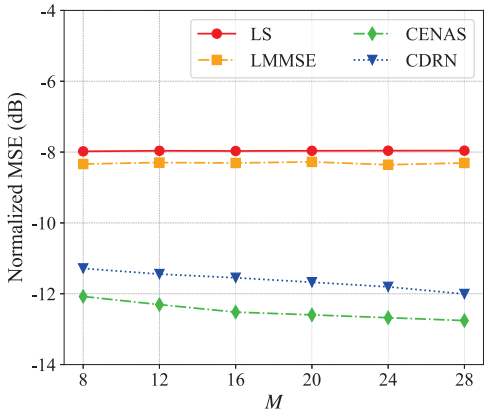


(b) The spectrum efficiency

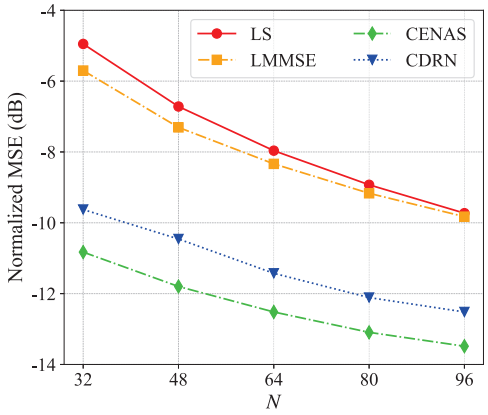
Fig. 7. (a) NMSE performance under $M = 16$, $N = 64$, and $C = 65$. (b) The spectrum efficiency based on different channel estimation results under $M = 16$, $N = 64$, $C = 65$ and SNR = 5.

conventional methods, suggests that the network architecture of CDRN struggles when dealing with very subtle noise. On the other hand, this figure demonstrates the stability and robustness of our algorithm, as our method can still outperform traditional methods in the presence of minor noise. Note that our proposed methodology leverages the automated generation of the sophisticated network architectures, resulting in an adaptive and highly effective neural network for the specific task. Moreover, in comparison to the manual design of networks, which often entails a time-consuming and iterative process relying on experiential trial and error, our method offers improved interpretability in network architecture design. The CENAS method also overperforms traditional approaches, such as the LS method. Particularly in low SNR scenarios, CENAS showcases a substantial performance gap of 12 dB when compared to the LS method. As the SNR increases, the performance gap gradually diminishes at SNR = 20. Furthermore, the LMMSE method which leverages statistical knowledge of the channel outperforms the LS method in low SNR conditions. However, there still remains a performance gap of approximately 4 dB when compared to CENAS. As the SNR increases, the performance gap between LMMSE and CENAS diminishes.

Obviously, the DL-methods show high performance in low



(a) Different numbers of elements in IRS



(b) Different numbers of antennas

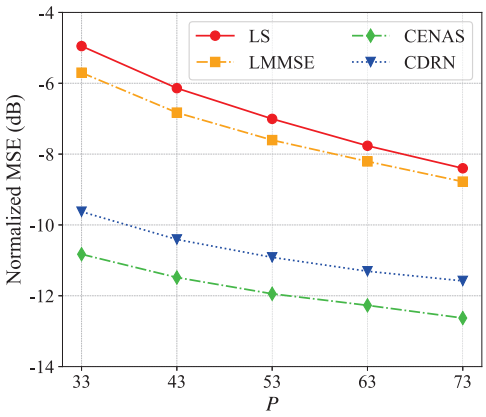
Fig. 8. NMSE performance with different numbers of N under M = 8, $C = N+1$, and SNR = 5 dB and different numbers of M under $N = 64, C = 65$, and SNR = 5 dB.

SNR scenarios, owing to their utilization of channel matrix data and non-linear mapping capabilities [23]. This disparity in network architectures between CENAS and CDRN [16] leads to divergent abilities in capturing and representing the intricate relationships within the channel data, ultimately impacting the performance of the respective methods.

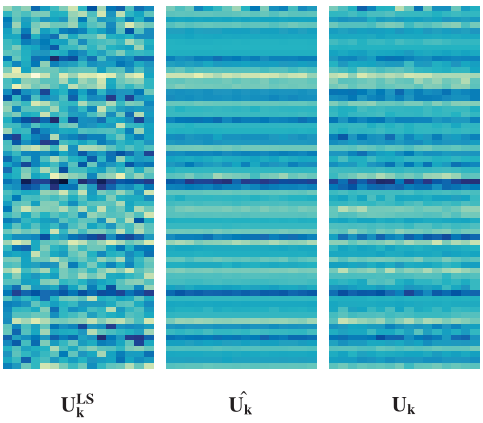
Fig. 7(b) illustrates the spectral efficiency based on different channel estimation methods. The standard maximum ratio transmission (MRT) precoding scheme with equal power allocation for all users is employed in our analysis. This observation shows that our proposed method exhibits the highest sum spectrum efficiency among the benchmark channel estimation methods considered. This outcome indicates that our method offers the most accurate estimation of the CSI compared to the alternative techniques.

D. Performance with Different Channel Dimensionalities and pilot numbers

To show the robustness of CENAS, we perform simulations to evaluate the estimation performance under varying channel dimensionality. In particular, we investigate the impact of two key parameters: the number of elements in the IRS denoted by N, and the number of antennas at the BS, denoted by M. These parameters directly influence the number of columns and rows in the channel, respectively. By systematically varying N and M, we gain insights into how changes in channel dimensionality affect the performance of our proposed method in accurately estimating the channel characteristics. In Fig. 8, the result provides compelling evidence regarding the adaptability and robustness of the neural network generated by the CENAS framework across various channel configurations.



(a) Different numbers of pilot



(b) Channel visualization

Fig. 9. (a) NMSE performance with different numbers of P under M = 16, N = 32, and SNR = 5 dB. (b) Heatmaps of the input channel $\mathbf{U_k^{LS}}$, the denoise output of CENAS $\hat{\mathbf{U}}_k^{LS}$, and the perfect CSI $\mathbf{U_k}$ under M = 16, $N = 64$, and SNR = 0 dB.

Furthermore, we conduct an analysis on the impact of pilot number on the NMSE performance. As depicted in Fig. 9(a), it is evident that all algorithms exhibit the improved performance as the number of pilots increases. This enhancement can be attributed to the higher quality input $\mathbf{U}_k^{LS}$ achieved through an increased number of pilots, thereby reducing the difficulty of the subsequent denoise process. The neural network architecture generated by CENAS method shows effective handling of input dimensional expansion and variations in noise, resulting in an enhanced accuracy of channel estimation with superior performance compared to alternative algorithms.

To provides a graphical depiction of the channel's characteristics, we present the heatmap of the channel in Fig. 9(b). The channel data are normalized to the range of $[-0.5, 0.5]$. Subsequently, both the input matrix $\mathbf{U}_k^{LS}$ and the output matrix of model D are transformed into heatmaps. The heatmap of the

initial input $\mathbf{U}_k^{LS}$ displays numerous points that are affected by noise. However, after undergoing the denoise process using model D, the resulting heatmap closely resembles the ideal CSI denoted by $\mathbf{U_k}$.

E. Performance with DeepMIMO Dataset

TABLE II
DEEPMIMO DATASET PARAMETERS

DeepMIMO Dataset parameter	Value
Frequency Band	28 GHz
Active BSs	3
Active Users (Receivers)	From Row R1000 to Row R1200
Active User (Transmitter)	Row R850, Column 90
Number of BS Antennas	(Mx, My, Mz)=(1, 64, 1)
Number of IRS Antennas	(Mx, My, Mz)=(1, 64, 1)
Antenna Spacing	0.5λ
System Bandwidth	100 MHz
OFDM Limit	1
Number of Paths	1

As a data-driven deep learning-based approach, it is imperative to conduct simulation using more realistic dataset to enhance the feasibility of our research. Recently, ray-tracing techniques are harnessed to generate channels, meticulously considering the influence of critical environmental factors.

Therefore, this experiment is conducted in the outdoor ray-tracing scenario denoted as 'O1' in the DeepMIMO dataset [37]. It is worth noting that we exclude the LOS path due to the dataset's lack of a scenario that aligns perfectly with our system model. Following the configuration detailed in [52], Table II succinctly outlines the pertinent parameters pertaining to this simulation.

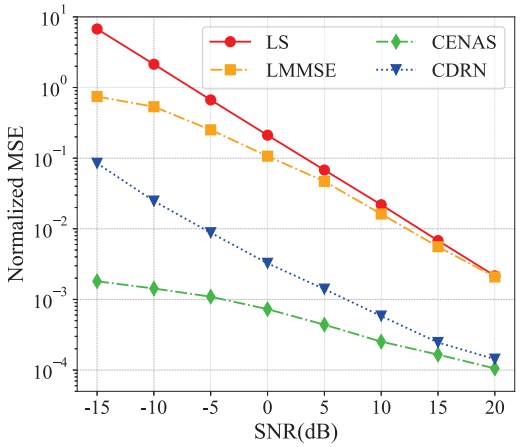


Fig. 10: NMSE performance under $M = 64$, $N = 64$, and $C = 64$.

Fig. 10 illustrates the NMSE perfomance on the DeepMIMO dataset. The CENAS curve stands out as the most exceptional, particularly within the low SNR range, signifying its superior noise reduction capabilities in challenging communication environments. Following closely is the CDRN [16], a deep learning-based approach that notably surpasses traditional mathematical methods. On the other hand, the LS method exhibits its limitations in the low SNR range, with its performance akin to random noise at SNR = -15 dB. LMMSE exhibits improved performance compared to LS, yet it remains markedly distant from the accuracy achieved by the deep learning-based methods, CDRN and CENAS.

V. Conclusion

We proposed the CENAS-based cascaded channel estimator for IRS aided multiuser MISO communication system. In contrast to the existing DL-based method, the CENAS saves the prohibited time and computation consumption of manually designing network architectures in a trial-and-error manner. Utilizing CENAS, the neural network architecture design reduces dependence on the expert's deep learning domain knowledge and eliminates the impediments to neural network applications in communications. Additionally, our method introduces an optimization balance with a well-initialized search strategy, enabling efficient and fair network exploration. By employing the truncated RAD algorithm, our approach ensures asymptotic convergence and mitigates the occurrence of skip collapse phenomena. Simulation results validate the superiority of the network constructed by CENAS, which obtains lower NMSE performance and provides the most accurate estimated CSI compared to other benchmark channel estimation methods.

Our future efforts will be dedicated to enhancing the interpretability of our method and extending its applicability to a wide range of communication tasks. This will be achieved by incorporating the emerging field of neural tangent kernel (NTK) theory [53], which will provide a more solid theoretical basis for designing neural networks specifically tailored for communication scenarios, rather than relying solely on empirical trial-and-error approaches.

Appendix A
Proof of Lemma 1 and Lemma 2

The proof of Lemma1 following the [38]. Let $\frac{\partial \mathcal{L}_{val}(\alpha)}{\partial \alpha} - \mathbf{g}_{T-M} = e_M$. According to the definition of $\mathbf{g}_{T-M}$, we have

$$\begin{aligned} e_M = &\left(\sum_{t=0}^{T-M} B_t A_{t+1} \cdots A_{T-M}\right) \\ &\cdot A_{T-M+1} \cdots A_T \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega}. \end{aligned} \tag{37}$$

Now, consider the case when $\mathcal{L}_{train}$ is locally α-strongly convex with respect to ω in the neighborhood of $\{\omega_{T-M-1}, \ldots, \omega_T\}$. We can then derive the following inequality

$$\begin{aligned} \|e_M\| \leq &\left\| \sum_{t=0}^{T-M} B_t A_{t+1} \cdots A_{T-M} \right\| \\ &\cdot \left\| A_{T-M+1} \cdots A_T \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega} \right\|, \end{aligned} \tag{38}$$

$$\begin{aligned} \|e_M\| \leq &(1-\gamma\alpha)^M \left\| \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega} \right\| \\ &\cdot \left\| \sum_{t=0}^{T-M} B_t A_{t+1} \cdots A_{T-M} \right\|. \end{aligned} \tag{39}$$

Suppose $\mathcal{L}_{train}$ is β-smooth but nonconvex. In the worst case scenario, if the smallest eigenvalue of $\frac{\partial^2 \mathcal{L}_{train}(\omega_{t-1},\alpha)}{\partial \boldsymbol{\omega}^2}$ is $-\beta$, then $\|A_t\| = 1+\gamma\beta \leq 2$ for $t = 0, \ldots, T-M$. Hence, (28) is proven.

The proof of Lemma2 following the [38]. If $\mathcal{L}_{train}$ is globally strongly convex, then we have

$$\begin{aligned}\|e_M\| \leq & \left\| \frac{\partial \mathcal{L}_{val}\left(\omega^*, \alpha\right)}{\partial \omega} \right\| (1-\gamma\alpha)^M \\ & \cdot \max_{t\in\{0,\ldots,T-M\}} \|B_t\| \sum_{t=0}^{T-M} (1-\gamma\alpha)^t.\end{aligned} \tag{40}$$

(29) is obtained by noting that

$$\sum_{t=0}^{T-M} (1-\gamma\alpha)^t \leq \sum_{t=0}^{\infty} (1-\gamma\alpha)^t = \frac{1}{\gamma\alpha}. \tag{41}$$

Let α_τ denote the τ-th iterate and F denote $\mathcal{L}_{val}$. For brevity, we use the notation $\nabla_\alpha F_\tau = \frac{\partial \mathcal{L}_{val}(\alpha_\tau)}{\partial \alpha_\tau}$ and $\mathbf{g}_{T-M,(\tau)} = \mathbf{g}_{T-M}\left(\alpha_\tau\right)$. Assuming that F is L-smooth, for almost all τ that $\|\nabla_\alpha F_\tau\| \leq G$ and $\left\|\mathbf{g}_{T-M,(\tau)}\right\| \leq G$ surely, we can employ L-smoothness to obtain the following inequality

$$\begin{aligned}F\left(\alpha_{\tau+1}\right) \leq & F\left(\alpha_\tau\right) + \left\langle \nabla F\left(\alpha_\tau\right), \alpha_{\tau+1} - \alpha_\tau \right\rangle \\ & + \frac{L}{2}\left\|\alpha_{\tau+1} - \alpha_\tau\right\|^2.\end{aligned} \tag{42}$$

Let $e_\tau = \nabla_\alpha F_\tau - \mathbf{g}_{T-M,(\tau)}$ represent the error in the gradient estimate. By substituting the recursive update $\alpha_{\tau+1} = \alpha_\tau - \eta_t \mathbf{g}_{T-M,(\tau)}$ into the inequality above, we can deduce the following inequality conditioned on α_τ in Eq(32) at the bottom of last page.

Using the inequalities Eq(33) and Eq(34) we can upper bound $\mathbb{E}_{|\alpha_\tau}\left[F\left(\alpha_{\tau+1}\right)\right]$ as Eq(35).

By performing a telescoping sum with the above inequality, we obtain Eq(36).

Dividing both sides by $\sum_{\tau=1}^{R} \eta_\tau$ and utilizing the facts that $\eta_\tau = O\left(\frac{1}{\sqrt{\tau}}\right)$ and $\frac{\sum_{\tau=1}^{R} \frac{1}{\tau}}{\sum_{\tau=1}^{R} \frac{1}{\sqrt{\tau}}} = O\left(\frac{\log R}{\sqrt{R}}\right)$, we can conclude the proof of the theorem.

REFERENCES

[1] H. Shi, T. Ji, Z. Zhang, L. Yang, and Y. Huang, "Automatic neural network construction-based channel estimation for irs-aided communication systems," in *2023 IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[2] C. Huang, A. Zappone, G. C. Alexandropoulos, M. Debbah, and C. Yuen, "Reconfigurable intelligent surfaces for energy efficiency in wireless communication," *IEEE Transactions on Wireless Communications*, vol. 18, no. 8, pp. 4157–4170, 2019.

[3] Q. Wu and R. Zhang, "Intelligent reflecting surface enhanced wireless network via joint active and passive beamforming," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5394–5409, 2019.

[4] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, "Intelligent reflecting surface-aided wireless communications: A tutorial," *IEEE Transactions on Communications*, vol. 69, no. 5, pp. 3313–3351, 2021.

[5] Z. Wang, L. Liu, S. Zhang, and S. Cui, "Massive mimo communication with intelligent reflecting surface," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2566–2582, 2023.

[6] J. Lin, G. Wang, R. Fan, T. A. Tsiftsis, and C. Tellambura, "Channel estimation for wireless communication systems assisted by large intelligent surfaces," *arXiv preprint arXiv:1911.02158*, 2019.

[7] M. Hua, Q. Wu, C. He, S. Ma, and W. Chen, "Joint active and passive beamforming design for irs-aided radar-communication," *IEEE Transactions on Wireless Communications*, vol. 22, no. 4, pp. 2278–2294, 2023.

[8] X. Guan, Q. Wu, and R. Zhang, "Anchor-assisted channel estimation for intelligent reflecting surface aided multiuser communication," *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3764–3778, 2022.

[9] M. Cui, G. Zhang, and R. Zhang, "Secure wireless communication via intelligent reflecting surface," *IEEE Wireless Communications Letters*, vol. 8, no. 5, pp. 1410–1414, 2019.

[10] Q. Wu and R. Zhang, "Joint active and passive beamforming optimization for intelligent reflecting surface assisted swipt under qos constraints," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1735–1748, 2020.

[11] ——, "Weighted sum power maximization for intelligent reflecting surface aided swipt," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 586–590, 2020.

$$\mathbb{E}_{|\alpha_\tau}\left[F\left(\alpha_{\tau+1}\right)\right] \leq F\left(\alpha_\tau\right) + \mathbb{E}_{|\alpha_\tau}\left[-\eta_t\left\langle \nabla F\left(\alpha_\tau\right), \mathbf{g}_{T-M,(\tau)}\right\rangle + \frac{L\eta_t^2}{2}\left\|\mathbf{g}_{T-M,(\tau)}\right\|^2\right]. \tag{32}$$

$$\begin{aligned}-\mathbb{E}_{|\alpha_\tau}\left[\left\langle \nabla F\left(\alpha_\tau\right), \mathbf{g}_{T-M,(\tau)}\right\rangle\right] &= \mathbb{E}_{|\alpha_\tau}\left[-\left\langle \nabla F\left(\alpha_\tau\right), \nabla_\alpha F_\tau\right\rangle + \left\langle \nabla F\left(\alpha_\tau\right), e_\tau\right\rangle\right] \\ &\leq -\left\|\nabla F\left(\alpha_\tau\right)\right\|^2 + G\left\|e_\tau\right\|.\end{aligned} \tag{33}$$

$$\frac{1}{2}\left\|\mathbf{g}_{T-M,(\tau)}\right\|^2 = \frac{1}{2}\left\|\nabla_\alpha F_\tau\right\|^2 + \frac{1}{2}\left\|e_\tau\right\|^2 - \left\langle \nabla_\alpha F_\tau, \mathbf{g}_{T-M,(\tau)}\right\rangle \leq \frac{3G^2}{2} + \frac{1}{2}\left\|e_\tau\right\|^2, \tag{34}$$

$$\mathbb{E}_{|\alpha_\tau}\left[F\left(\alpha_{\tau+1}\right)\right] \leq F\left(\alpha_\tau\right) + \mathbb{E}_{|\alpha_\tau}\left[-\eta_\tau\left\|\nabla F\left(\alpha_\tau\right)\right\|^2 + \eta_\tau G\left\|e_\tau\right\| + L\eta_\tau^2\left(\frac{3G^2}{2} + \frac{1}{2}\left\|e_\tau\right\|^2\right)\right]. \tag{35}$$

$$\begin{aligned}\mathbb{E}\left[\sum_{\tau=1}^{R} \eta_\tau\left\|\nabla F\left(\alpha_\tau\right)\right\|^2\right] &\leq F\left(\alpha_1\right) + \mathbb{E}\left[\sum_{\tau=1}^{R} G\eta_\tau\left\|e_\tau\right\| + L\eta_\tau^2\left(\frac{3G^2}{2} + \frac{1}{2}\left\|e_\tau\right\|^2\right)\right] \\ &\leq F\left(\alpha_1\right) + \sum_{\tau=1}^{R}\left(G\epsilon\eta_\tau + \frac{L\left(3G^2+\epsilon^2\right)}{2}\eta_\tau^2\right).\end{aligned} \tag{36}$$

[12] X. Guan, Q. Wu, and R. Zhang, "Joint power control and passive beamforming in irs-assisted spectrum sharing," *IEEE Communications Letters*, vol. 24, no. 7, pp. 1553–1557, 2020.

[13] B. Zheng, Q. Wu, and R. Zhang, "Intelligent reflecting surface-assisted multiple access with user pairing: Noma or oma?" *IEEE Communications Letters*, vol. 24, no. 4, pp. 753–757, 2020.

[14] W. Xie, J. Xiao, P. Zhu, C. Yu, and L. Yang, "Deep compressed sensing-based cascaded channel estimation for ris-aided communication systems," *IEEE Wireless Communications Letters*, 2022.

[15] G. Zhou, C. Pan, H. Ren, P. Popovski, and A. L. Swindlehurst, "Channel estimation for ris-aided multiuser millimeter-wave systems," *IEEE Transactions on Signal Processing*, 2022.

[16] C. Liu, X. Liu, D. W. K. Ng, and J. Yuan, "Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications," *IEEE Transactions on Wireless Communications*, 2021.

[17] T. L. Jensen and E. De Carvalho, "An optimal channel estimation scheme for intelligent reflecting surfaces based on a minimum variance unbiased estimator," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 5000–5004.

[18] J. Chen, Y.-C. Liang, H. V. Cheng, and W. Yu, "Channel estimation for reconfigurable intelligent surface aided multi-user mimo systems," *arXiv preprint arXiv:1912.03619*, 2019.

[19] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Communications Letters*, vol. 23, no. 4, pp. 652–655, 2019.

[20] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep cnn-based channel estimation for mmwave massive mimo systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 989–1000, 2019.

[21] M. B. Mashhadi and D. Gündüz, "Pruning the pilots: Deep learning-based pilot design and channel estimation for mimo-ofdm systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 10, pp. 6315–6328, 2021.

[22] Y. Jin, J. Zhang, C. Huang, L. Yang, H. Xiao, B. Ai, and Z. Wang, "Multiple residual dense networks for reconfigurable intelligent surfaces cascaded channel estimation," *IEEE Transactions on Vehicular Technology*, 2021.

[23] Z. Zhang, T. Ji, H. Shi, C. Li, Y. Huang, and L. Yang, "A self-supervised learning-based channel estimation for irs-aided communication without ground truth," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer, 2015, pp. 234–241.

[26] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Advances in neural information processing systems*, vol. 24, 2011.

[28] S. Shen, Y. Cheng, Z. He, W. He, H. Wu, M. Sun, and Y. Liu, "Minimum risk training for neural machine translation," *arXiv preprint arXiv:1512.02433*, 2015.

[29] S. Thrun and L. Pratt, *Learning to learn*. Springer Science & Business Media, 2012.

[30] R. Yang, Z. Zhang, X. Zhang, C. Li, Y. Huang, and L. Yang, "Meta-learning for beam prediction in a dual-band communication system," *IEEE Transactions on Communications*, vol. 71, no. 1, pp. 145–157, 2023.

[31] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International conference on machine learning*. PMLR, 2018, pp. 4095–4104.

[32] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[33] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer, "Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 734–10 742.

[34] H. Cai, L. Zhu, and S. Han, "Proxylessnas: Direct neural architecture search on target task and hardware," *arXiv preprint arXiv:1812.00332*, 2018.

[35] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[36] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in *Proceedings of the aaai conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 4780–4789.

[37] A. Alkhateeb, "DeepMIMO: A generic deep learning dataset for millimeter wave and massive MIMO applications," in *Proc. of Information Theory and Applications Workshop (ITA)*, San Diego, CA, Feb 2019, pp. 1–8.

[38] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 1723–1732.

[39] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[41] H. Zhang, Y. Li, H. Chen, and C. Shen, "Memory-efficient hierarchical neural architecture search for image denoising," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 3657–3666.

[42] F. Pedregosa, "Hyperparameter optimization with approximate gradient," in *International conference on machine learning*. PMLR, 2016, pp. 737–746.

[43] L. Franceschi, M. Donini, P. Frasconi, and M. Pontil, "Forward and reverse gradient-based hyperparameter optimization," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1165–1173.

[44] R. Liu, J. Gao, J. Zhang, D. Meng, and Z. Lin, "Investigating bi-level optimization for learning and vision from a unified perspective: A survey and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 12, pp. 10 045–10 067, 2022.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[46] H. Pham, M. Guan, B. Zoph, Q. Le, and J. Dean, "Efficient neural architecture search via parameters sharing," in *International conference on machine learning*. PMLR, 2018, pp. 4095–4104.

[47] X. Chu, X. Wang, B. Zhang, S. Lu, X. Wei, and J. Yan, "Darts-: robustly stepping out of performance collapse without indicators," *arXiv preprint arXiv:2009.01027*, 2020.

[48] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016.

[49] P. Mishra and K. Sarawadekar, "Polynomial learning rate policy with warm restart for deep neural network," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 2087–2092.

[50] R. Wang, M. Cheng, X. Chen, X. Tang, and C.-J. Hsieh, "Rethinking architecture selection in differentiable nas," *arXiv preprint arXiv:2108.04392*, 2021.

[51] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," *Advances in neural information processing systems*, vol. 31, 2018.

[52] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Enabling large intelligent surfaces with compressive sensing and deep learning," *IEEE Access*, vol. 9, pp. 44 304–44 321, 2021.

[53] L. Xiao, J. Pennington, and S. Schoenholz, "Disentangling trainability and generalization in deep neural networks," in *International Conference on Machine Learning*. PMLR, 2020, pp. 10 462–10 472.

This article has been accepted for publication in IEEE Transactions on Wireless Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TWC.2024.3374352

Haoqing Shi (Student Member, IEEE) received the bachelor's degree in Electronic and Information Engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2017. He is currently pursuing the Ph.D. degree in information and communication engineering with the School of Information Science and Engineering, Southeast University. His research interests mainly focus on intelligent wireless communications.

Yongming Huang (Senior Member, IEEE) received the B.S. and M.S. degrees from Nanjing University, Nanjing, China, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 2007.

Since March 2007 he has been a faculty in the School of Information Science and Engineering, Southeast University, China, where he is currently a full professor. He has also been the Director of the Pervasive Communication Research Center, Purple Mountain Laboratories, since 2019. From 2008 to 2009, he was visiting the Signal Processing Lab, Royal Institute of Technology (KTH), Stockholm, Sweden. He has published over 200 peer-reviewed papers, hold over 80 invention patents. His current research interests include intelligent 5G/6G mobile communications and millimeter wave wireless communications. He submitted around 20 technical contributions to IEEE standards, and was awarded a certificate of appreciation for outstanding contribution to the development of IEEE standard 802.11aj. He has served as an Associate Editor for a couple of international journals including the IEEE Transactions on Signal Processing, the IEEE Wireless Communications Letters.

Shi Jin (Fellow, IEEE) received the B.S. degree in communications engineering from the Guilin University of Electronic Technology, Guilin, China, in 1996, the M.S. degree from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2003, and the Ph.D. degree in information and communications engineering from Southeast University, Nanjing, in 2007. From June 2007 to October 2009, he was a Research Fellow with University College London, Adastral Park Research Campus, London, U.K. He is currently with the Faculty of the National Mobile Communications Research Laboratory, Southeast University. His research interests include space time wireless communications, random matrix theory, and information theory. He and his coauthors have been awarded the 2011 IEEE Communications Society Stephen O. Rice Prize Paper Award in the field of communication theory and the 2010 Young Author Best Paper Award by the IEEE Signal Processing Society. He served as an Associate Editor for the IEEE Transactions on Wireless Communications, IEEE Communications Letters, and IET Communications.

Zheng Wang (Senior Member, IEEE) received the B.S. degree in electronic and information engineering from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2009, and the M.S. degree in communications from University of Manchester, Manchester, U.K., in 2010. He received the Ph.D degree in communication engineering from Imperial College London, UK, in 2015.

Since 2021, he has been an Associate Professor in the School of Information and Engineering, Southeast University (SEU), Nanjing, China. From 2015 to 2016 he served as a Research Associate at Imperial College London, UK. From 2016 to 2017 he was an senior engineer with Radio Access Network R&D division, Huawei Technologies Co.. From 2017 to 2020 he was an Associate Professor at the College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China. His current research interests include massive MIMO systems, machine learning and data analytics over wireless networks, and lattice theory for wireless communications.

Luxi Yang (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 1990 and 1993, respectively. Since 1993, he has been with the Department of Radio Engineering, Southeast University, where he is currently a Full Professor of Information Systems and Communications, and the Director of the Digital Signal Processing Division. He has authored or coauthored of two published books and more than 200 journal papers and holds 50 patents. His current research interests include signal processing for wireless communications, MIMO communications, intelligent wireless communications, and statistical signal processing. Prof. Yang received the first- and second-class prizes of science and technology progress awards of the State Education Ministry of China in 1998, 2002, and 2014. He is currently a member of Signal Processing Committee of the Chinese Institute of Electronics.