This article has been accepted for publication in IEEE Wireless Communications Letters. 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/LWC.2025.3584268

An SST-IE Framework for Beamforming and Phase Shift Design in RIS-aided Multi-user Networks

Xiang Huang, Joohyun Cho and Rong-Rong Chen

***Abstract*—This paper proposes a novel semi-supervised, separate training, and iterative execution (SST-IE) framework for joint beamforming and phase shift optimization in RIS-aided multiuser networks. SST-IE employs three independently trained neural networks with carefully designed input features and loss functions, effectively leveraging the strengths of both supervised and unsupervised learning. Guided by the block coordinate descent (BCD) algorithm, SST-IE achieves superior sum-rate performance compared to the BCD algorithm while requiring significantly fewer iterations. Simulation results show that SST-IE outperforms representative deep learning and reinforcement learning (RL) approaches—including a two-stage method and a Deep Deterministic Policy Gradient (DDPG)-based RL algorithm—particularly in high-SNR regimes, while ensuring fair rate allocation among users.**

***Index Terms*—Reconfigurable intelligent surface (RIS), beamforming, phase shift, semi-supervised learning, block-coordinate descent (BCD).**

I. Introduction

As wireless systems move to higher frequencies, signals face increased blockage and attenuation. Reconfigurable intelligent surfaces (RIS) have emerged as a promising solution [1]. Comprising passive elements that adjust the phase of incident waves, a RIS can steer reflections to align multipaths constructively at the receiver, improving signal strength and mitigating propagation loss.

In a RIS-aided single-cell multiuser network, jointly optimizing the base station (BS) beamforming matrix $\mathbf{W}$ and RIS phase shifts $\boldsymbol{\theta}$ is challenging due to their coupled effect on the sum-rate. Given fixed RIS phases, optimizing $\mathbf{W}$ reduces to a multiuser MISO precoding problem, for which the Weighted Minimum Mean Square Error (WMMSE) method is a standard solution [2]. A low-complexity block coordinate descent (BCD) algorithm is proposed in [3] that achieves performance comparable to alternating optimization (AO) with WMMSE. This BCD method has since been widely adopted as a baseline in works such as [4], [5]. While effective, BCD typically requires many iterations to converge, making it computationally intensive.

In addition to the conventional optimization approaches discussed above, recent studies have explored machine learning methods to address the challenges of joint beamforming and phase shift design [6]–[8]. A common framework, introduced in [6], employs a two-stage neural network—PhaseNet for $\boldsymbol{\theta}$ and BeamNet for $\mathbf{W}$—jointly trained via unsupervised learning to maximize sum-rate. This architecture has been extended to time-varying channels [7] and hybrid beamforming [8]. A Transformer-based two-stage design [9] and reinforcement learning methods, such as Deep Deterministic Policy Gradient (DDPG) [10], have also been explored for RIS control.

The authors are with the Dept. of ECE, University of Utah, USA. The work is supported in part by NSF grants CNS-2229562 and ECCS-2153875.

In this work, we propose a novel neural network framework for joint beamforming and phase shift design, termed semi-supervised, separate training, and iterative execution (SST-IE). The key innovations of the proposed approach are summarized below:

- The proposed SST-IE framework employs separate training of three neural networks: PhaseNet (PN), BeamNet (BN), and fine-tuning BeamNet (FBN). This separation, combined with semi-supervised learning, is key to SST-IE's effectiveness. To enable independent training, we introduce novel inputs—namely, the symbol-level cascaded (or effective) channel—for PN and FBN. This design not only simplifies training but also enables efficient iterative execution during inference.
- The SST-IE framework leverages semi-supervised learning with newly designed loss functions to achieve two key objectives: maximizing the sum-rate via unsupervised learning and utilizing BCD-generated solutions through supervised learning. The latter is realized by guiding PN to produce an effective channel matrix that aligns with the BCD output, and by training BN to match its beamforming matrix to that derived from BCD.
- Through iterative execution, SST-IE achieves sum-rate performance that surpasses the BCD algorithm with significantly fewer iterations. It also outperforms both the two-stage deep learning method and the RL-DDPG approach, while overcoming their single-user transmission limitation and enabling fair rate allocation across all users.

Notation: The scalar, vector, and matrix are lowercase, bold lowercase, and bold uppercase, i.e., a, $\mathbf{a}$, and $\mathbf{A}$, respectively. The transpose and conjugate transpose operators are denoted as $(\cdot)^t$, $(\cdot)^*$. The circularly symmetric complex Gaussian distribution with mean μ and variance σ^2 is denoted as $\mathcal{CN}(\mu, \sigma^2)$.

II. Problem Setup

In this section, we describe the basic problem setup. We begin with a description of the system model, followed by beamforming and phase-shift design.

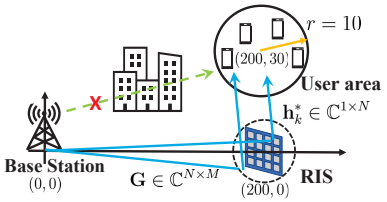


Fig. 1: A RIS-aided single-cell multiuser Network.

A. System model

As shown in Fig. 1, the system comprises a BS with M antennas, K single-antenna users, and a RIS with N unit-cells.

Authorized licensed use limited to: The University of Utah. Downloaded on July 25,2025 at 17:04:57 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Wireless Communications Letters. 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/LWC.2025.3584268

The signal s_l from BS to user l is beamformed by the vector $\mathbf{w}_l \in \mathbb{C}^{M\times 1}$. The beamforming matrix $\mathbf{W} = [\mathbf{w}_1, \cdots, \mathbf{w}_K] \in \mathbb{C}^{M\times K}$. The channel from BS to RIS is $\mathbf{G} \in \mathbb{C}^{N\times M}$, and from RIS to user k is $\mathbf{h}_k^* \in \mathbb{C}^{1\times N}$. We consider a narrowband channel model [3]. The channels $\mathbf{G}$ and $\mathbf{h}_k$ can be written as

$$\begin{aligned}\mathbf{G} &= L_1\big(\sqrt{\varepsilon/(\varepsilon+1)}\,\mathbf{a}_N(\vartheta)\mathbf{a}_M(\varphi)^* + \sqrt{1/(\varepsilon+1)}\,\bar{\mathbf{G}}\big)\\ \mathbf{h}_k &= L_{2,k}\big(\sqrt{\varepsilon/(\varepsilon+1)}\,\mathbf{a}_N(\xi_k) + \sqrt{1/(\varepsilon+1)}\,\bar{\mathbf{h}}_k\big),\end{aligned} \tag{1}$$

where L_1 and $L_{2,k}$ are the path gains from the BS to RIS and from the RIS to user k, respectively. $\varepsilon = 10$ is the Rician factor. $\bar{\mathbf{G}}$ and $\bar{\mathbf{h}}_k$ denote the non-line-of-sight (NLOS) components with elements drawn from $\mathcal{CN}(0,1)$. Let $\mathbf{a}_N(\vartheta)$ and $\mathbf{a}_N(\xi_k)$ denote the RIS receive and transmit steering vectors, and $\mathbf{a}_M(\varphi)$ the BS transmit steering vector. Here, ϑ (ξ_k) represents the angle of arrival (departure) at the RIS, and φ the angle of departure at the BS. Since uniform linear arrays (ULAs) are assumed at both the BS and RIS, the steering vector with L antennas is defined as $\mathbf{a}_L(\vartheta) = \frac{1}{\sqrt{L}}\big[1;\cdots;e^{j\pi(L-1)\sin(\vartheta)}\big] \in \mathbb{C}^{L\times 1}$. The RIS phase shift matrix is defined as $\mathbf{\Theta} = \mathrm{diag}(\theta_1, ..., \theta_N) \in \mathbb{C}^{N\times N}$, where $\theta_r = e^{j\psi_r}, r \in \{1, 2, ..., N\}$, and ψ_r is the steering direction of the r-th RIS element. The received signal at user k is

$$y_k = \mathbf{h}_k^*\mathbf{\Theta}\mathbf{G}\sum_{l=1}^{K}\mathbf{w}_l s_l + n_k = \boldsymbol{\theta}^*\mathrm{diag}(\mathbf{h}_k^*)\mathbf{G}\sum_{l=1}^{K}\mathbf{w}_l s_l + n_k,$$

where $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_N]^* \in \mathbb{C}^{N\times 1}$ and the noise $n_k \sim \mathcal{CN}(0, \sigma_k^2)$. The cascaded channel for user k is defined as $\mathbf{H}_k = \mathrm{diag}(\mathbf{h}_k^*)\mathbf{G} \in \mathbb{C}^{N\times M}$. The effective channel between BS and user k via RIS is defined as $\mathbf{f}_k^* = \mathbf{h}_k^*\mathbf{\Theta}\mathbf{G} = \boldsymbol{\theta}^*\mathbf{H}_k \in \mathbb{C}^{1\times M}$. We also define $\mathbf{F} = [\mathbf{f}_1^*; \cdots; \mathbf{f}_K^*] \in \mathbb{C}^{K\times M}$ as the combined effective channel matrix.

B. Beamforming and phase-shift design

Our goal is to optimize $\mathbf{W}$ and $\mathbf{\Theta}$ so that the sum-rate is maximized. The sum-rate maximization is formulated as

$$\begin{aligned}&\max_{\mathbf{W},\boldsymbol{\theta}} f(\mathbf{W}, \boldsymbol{\theta}) = \sum_{k=1}^{K}\log(1+\gamma_k)\\ &\quad \text{s.t. } |\theta_n| = 1, \forall n = 1, \ldots, N, \quad \sum_{l=1}^{K}|\mathbf{w}_l|^2 \le P_T,\end{aligned} \tag{2}$$

where P_T is BS's transmit power. We denote γ_k as the signal to interference plus noise ratio (SINR) of user k, given by $\gamma_k = |\mathbf{f}_k^*\mathbf{w}_k|^2 / \big(\sum_{l=1,l\neq k}^{K}|\mathbf{f}_k^*\mathbf{w}_l|^2 + \sigma_k^2\big)$.

III. The Proposed SST-IE

In this section, we first detail the semi-supervised separate training strategy for the three key components—PN, BN, and FBN—of the proposed SST-IE design, highlighting how it addresses the limitations of the two-stage method and ensures fair user rate allocation. We then describe the iterative execution process and introduce a permutation enhancement technique to further improve performance.

A. Semi-supervised separate training (SST)

Given a channel realization and assuming the cascaded channels $\{\mathbf{H}_k\}_{k=1}^K$ are known, we first prepare the BCD-generated solutions [3] as training data for supervised training purpose. These solutions are denoted as $\{(\mathbf{W}_{\mathrm{BCD}}^{(j)}, \boldsymbol{\theta}_{\mathrm{BCD}}^{(j)}) | j \in \boldsymbol{J} = \{0, 25, 100, 500\}\}$, where each pair corresponds to the outputs at the j-th iteration of the BCD algorithm. The initial beamforming matrix $\mathbf{W}_{\mathrm{BCD}}^{(0)} = \mathbf{W}_{ini}$ is computed using the combined effective channel matrix $\mathbf{F}_{ini}$, which is constructed from a randomly generated phase shift $\boldsymbol{\theta}_{rnd}$ as

$$\begin{aligned}\mathbf{F}_{ini} &= [\boldsymbol{\theta}_{rnd}^*\mathbf{H}_1; \cdots; \boldsymbol{\theta}_{rnd}^*\mathbf{H}_K],\\ \mathbf{W}_{ini} &= \mathbf{F}_{ini}^*(\mathbf{F}_{ini}\mathbf{F}_{ini}^*)^{-1}.\end{aligned} \tag{3}$$

After normalizing $\mathbf{W}_{ini}$ to satisfy the power constraint, we run one iteration of BCD algorithm to obtain $\boldsymbol{\theta}_{\mathrm{BCD}}^{(0)}$. The remaining solution pairs are generated through subsequent iterations of the BCD algorithm. For brevity, we omit the explicit dependency of these solution pairs on specific channel realizations in the following discussion.

The proposed SST leverages BCD solutions to guide supervised training, which operates alongside an unsupervised learning objective aimed at maximizing the sum-rate. This combination of supervised guidance and unsupervised optimization enables SST-IE to achieve performance that can even surpass the BCD algorithm itself.

1) Training of PN

During the separate training for PN, a pre-defined beamforming matrix $\mathbf{W}$ is required. Here, we select $\mathbf{W}$ across the BCD solutions $\{\mathbf{W}_{\mathrm{BCD}}^{(j)}, j \in \boldsymbol{J}\}$, and introduce a new input, $\big[\mathbf{H}_1\mathbf{W}|\cdots|\mathbf{H}_K\mathbf{W}\big] \in \mathbb{C}^{K\times N\times K}$, where each $\mathbf{H}_k\mathbf{W} \in \mathbb{C}^{N\times K}$ represents the symbol-level cascaded channel for user k, and $[\cdots|\cdots|\cdots]$ denotes the depth-wise stacking. In $\mathbf{H}_k\mathbf{W} = [\mathbf{H}_k\mathbf{w}_1, \cdots, \mathbf{H}_k\mathbf{w}_K]$, each $\mathbf{H}_k\mathbf{w}_l \in \mathbb{C}^{N\times 1}$ represents the aggregated vector channel from symbol s_l to user k via RIS. Specifically, the n-th element of $\mathbf{H}_k\mathbf{w}_k$ corresponds to the desired signal path from s_k to user k through the n-th RIS element, while that of $\mathbf{H}_k\mathbf{w}_l$ ($l \neq k$) corresponds to interference from s_l.

In contrast, PhaseNet in the two-stage design [6] does not account for beamforming and uses only the cascaded channels $\{\mathbf{H}_k\}_{k=1}^K$ as input. The symbol-level cascaded channels $\{\mathbf{H}_k\mathbf{W}\}_{k=1}^K$ introduced here incorporate the beamforming matrix, making this a key distinction between the two-stage design and the proposed SST-IE. Moreover, our proposed PN input is a three-dimensional (3D) feature, while the input in [6] is two-dimensional (2D).

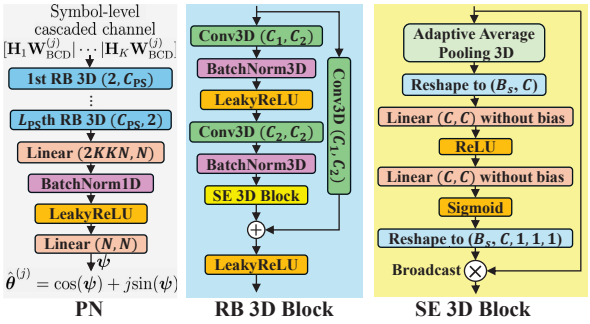


Fig. 2: Network structure of PN. (In the RB 3D block, the output of the SE 3D block is added to the RB 3D input via a skip connection. If $C_1 = C_2$, the input is directly added to the output; otherwise, it is first transformed to match C_2 before addition. The sum is then passed through LeakyReLU activation.)

As shown in Fig. 2, the input to PN is a 3D matrix of size $K \times N \times K$. PN consists of L_{PS} 3D residual blocks (RBs) followed by two fully connected (FC) layers. The first RB has

Authorized licensed use limited to: The University of Utah. Downloaded on July 25,2025 at 17:04:57 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Wireless Communications Letters. 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/LWC.2025.3584268

two input channels, corresponding to the real and imaginary parts. Each of the first $L_{\text{PS}}-1$ RBs has C_{PS} output channels, while the final RB has 2 output channels. The final FC layer produces N outputs, representing the RIS phase angles $\boldsymbol{\psi}$. Each 3D RB consists of two 3D convolutional (Conv3D) layers. Let C_1 and C_2 denote the number of input and output channels, respectively. The second Conv3D layer is followed by a 3D Bnorm layer and a squeeze-and-excitation (SE) 3D block, as described in [11].

With the symbol-level cascaded channels corresponding to $\mathbf{W}_{\text{BCD}}^{(j)}$ as input to the PN, the output is denoted by $\hat{\boldsymbol{\theta}}^{(j)}$. The loss $\mathcal{L}_1$ in (4), with weights w_1 and w_2, consists of two terms. The first term, $f(\mathbf{W}_{\text{BCD}}^{(j)}, \hat{\boldsymbol{\theta}}^{(j)})$, corresponds to unsupervised learning, aiming to maximize the sum-rate achieved by using $\mathbf{W}_{\text{BCD}}^{(j)}$ and the phase shift generated by the PN, $\hat{\boldsymbol{\theta}}^{(j)}$. The second term is for supervised learning, aiming to minimize the the normalized mean square error (NMSE) between the $\hat{\mathbf{F}}^{(j)}$ and $\mathbf{F}_{\text{BCD}}^{(j)}$, which are the combined effective channel matrix calculated from the PN output $\hat{\boldsymbol{\theta}}^{(j)}$ and the BCD solution $\boldsymbol{\theta}_{\text{BCD}}^{(j)}$, respectively.

$$\mathcal{L}_1 = -\frac{w_1}{|\boldsymbol{J}|}\sum_{j\in\boldsymbol{J}} \underbrace{f(\mathbf{W}_{\text{BCD}}^{(j)}, \hat{\boldsymbol{\theta}}^{(j)})}_{\text{unsupervised}} + \frac{w_2}{|\boldsymbol{J}|}\sum_{j\in\boldsymbol{J}} \underbrace{\text{NMSE}(\hat{\mathbf{F}}^{(j)}, \mathbf{F}_{\text{BCD}}^{(j)})}_{\text{supervised}} \quad (4)$$

2) Training of BN and FBN

During the Separate training of BN, a pre-determined phase shift $\boldsymbol{\theta}$ is required. Here, we select $\boldsymbol{\theta}$ across the BCD solutions $\{\boldsymbol{\theta}_{\text{BCD}}^{(j)}, j \in \boldsymbol{J}\}$. The input to BN is $[\boldsymbol{\theta}^*\mathbf{H}_1; \cdots; \boldsymbol{\theta}^*\mathbf{H}_K] \in \mathbb{C}^{K\times M}$, where each row $\boldsymbol{\theta}^*\mathbf{H}_k \in \mathbb{C}^{1\times M}$ is the effective channel for user k. Here, $[\cdots;\cdots;\cdots]$ denotes row-wise concatenation. The output is $\hat{\mathbf{W}}^{(j)}$. The loss $\mathcal{L}_2$ in (5), weighted by γ_1 and γ_2, comprises an unsupervised term and a supervised term: the sum-rate objective $f(\hat{\mathbf{W}}^{(j)}, \boldsymbol{\theta}_{\text{BCD}}^{(j)})$ and the NMSE between $\hat{\mathbf{W}}^{(j)}$ and $\mathbf{W}_{\text{BCD}}^{(j)}$,

$$\mathcal{L}_2 = -\frac{\gamma_1}{|\boldsymbol{J}|}\sum_{j\in\boldsymbol{J}} \underbrace{f(\hat{\mathbf{W}}^{(j)}, \boldsymbol{\theta}_{\text{BCD}}^{(j)})}_{\text{unsupervised}} + \frac{\gamma_2}{|\boldsymbol{J}|}\sum_{j\in\boldsymbol{J}} \underbrace{\text{NMSE}(\hat{\mathbf{W}}^{(j)}, \mathbf{W}_{\text{BCD}}^{(j)})}_{\text{supervised}} \quad (5)$$

Once BN has already been well trained, we proceed with the separate training of FBN. We select a pair $(\boldsymbol{\theta}, \mathbf{W})$ from $\{(\boldsymbol{\theta}_{\text{BCD}}^{(j)}, \hat{\mathbf{W}}^{(j)}), j \in \boldsymbol{J}\}$, where $\hat{\mathbf{W}}^{(j)}$ is the output from the trained BN given the input derived from $\boldsymbol{\theta}_{\text{BCD}}^{(j)}$. For each such pair $(\boldsymbol{\theta}_{\text{BCD}}^{(j)}, \hat{\mathbf{W}}^{(j)})$, the input feature to FBN is defined as $[\boldsymbol{\theta}_{\text{BCD}}^{(j)^*}\mathbf{H}_1\hat{\mathbf{W}}^{(j)}; \cdots; \boldsymbol{\theta}_{\text{BCD}}^{(j)^*}\mathbf{H}_K\hat{\mathbf{W}}^{(j)}] \in \mathbb{C}^{K\times K}$, where $\boldsymbol{\theta}_{\text{BCD}}^{(j)^*}\mathbf{H}_k\hat{\mathbf{W}}^{(j)}$ is referred to as the symbol-level effective channel for user k. These quantities depend on the beamforming matrix and are therefore distinct from the traditional effective channels. The corresponding output of FBN is denoted by $\hat{\mathbf{W}}_{\text{F}}^{(j)}$. The loss function $\mathcal{L}_3$, defined in (6) and weighted by α_1, consists of a single unsupervised term

$$\mathcal{L}_3 = -\frac{\alpha_1}{|\boldsymbol{J}|}\sum_{j\in\boldsymbol{J}} \underbrace{f(\hat{\mathbf{W}}_{\text{F}}^{(j)}, \boldsymbol{\theta}_{\text{BCD}}^{(j)})}_{\text{unsupervised}}. \quad (6)$$

FBN is designed to refine $\hat{\mathbf{W}}^{(j)}$ generated by BN. It consists of L_{F} sub-networks, where the i-th sub-network produces an increment $\triangle\mathbf{W}_i^{(j)}$ used to update the beamforming matrix as $\hat{\mathbf{W}}_i^{(j)} = \hat{\mathbf{W}}_{i-1}^{(j)} + \eta\triangle\mathbf{W}_i^{(j)}$ with η set to 0.01. The process is initialized with $\hat{\mathbf{W}}_0^{(j)} = \hat{\mathbf{W}}^{(j)}$ and the final output of FBN is given by $\hat{\mathbf{W}}_{\text{F}}^{(j)} = \hat{\mathbf{W}}_{L_{\text{F}}}^{(j)}$.

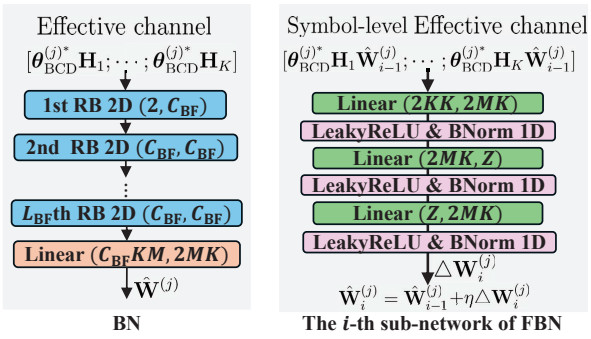


Fig. 3: Network structure of BN and the i-th FBN sub-network. (The 2D RB shares the same structure as the 3D RB, with minor adjustments to accommodate 2D dimensions.)

As shown in Fig. 3, the input to BN is a 2D matrix of size $K \times M$. BN consists of L_{BF} 2D RBs and one FC layer. Each RB has C_{BF} output channels and the FC layer produces $2MK$ outputs representing the real and imaginary parts of $\mathbf{W} \in \mathbb{C}^{M\times K}$. FBN comprises L_{F} sub-networks, each sharing the same structure, as shown in Fig. (3). The input to the i-th FBN sub-network is a 2D matrix of size $K \times K$. For the first sub-network, the input is the symbol-level effective channel $[\boldsymbol{\theta}_{\text{BCD}}^{(j)^*}\mathbf{H}_1\hat{\mathbf{W}}^{(j)}; \cdots; \boldsymbol{\theta}_{\text{BCD}}^{(j)^*}\mathbf{H}_K\hat{\mathbf{W}}^{(j)}]$, where $\hat{\mathbf{W}}_0^{(j)} = \hat{\mathbf{W}}^{(j)}$.

B. Ensure fair rate allocation via SST

We observe that the BCD algorithm [3] designs the phase shifts to shape the combined effective channel matrix $\mathbf{F} = [\mathbf{f}_1^*; \cdots; \mathbf{f}_K^*] \in \mathbb{C}^{K\times M}$ such that one row—corresponding to some user k—dominates with a significantly higher effective channel gain $\|\mathbf{f}_k\|^2$ than the others. The algorithm then allocates the least power to this user in its beamforming design, assigning more power to users with weaker effective channels. This strategy promotes a more balanced distribution of individual transmission rates across users.

In contrast, the PhaseNet in the two-stage approach of [6] also produces a dominant row in $\mathbf{F}$, but its position remains fixed regardless of the strength of the cascaded channels $\{\mathbf{H}_k\}_{k=1}^K$. Moreover, the two-stage method allocates all power to this fixed user, restricting its ability to support only the single user with the strongest effective channel.

The proposed SST framework employs semi-supervised learning to develop a power allocation strategy inspired by the behavior of the BCD algorithm. During PN training, the second term in the loss function $\mathcal{L}_1$ (see (4)) encourages alignment between the learned combined effective channel matrix $\hat{\mathbf{F}}^{(j)}$ and the BCD counterpart $\mathbf{F}_{\text{BCD}}^{(j)}$, capturing the implicit principle BCD uses to select the dominant row. For BN training, the loss $\mathcal{L}_2$ (see (5)) incorporates an NMSE term between $\hat{\mathbf{W}}^{(j)}$ and $\mathbf{W}_{\text{BCD}}^{(j)}$ to encourage balanced power allocation across users. Rather than imitating BCD, these components guide SST to learn more effective strategies, leading to improved sum-rate performance and fairer user rate distribution.

C. Iterative execution

During evaluation, we adopt an iterative framework that sequentially incorporates PN, BN, and FBN. In contrast, the

Authorized licensed use limited to: The University of Utah. Downloaded on July 25,2025 at 17:04:57 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Wireless Communications Letters. 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/LWC.2025.3584268

TABLE I: Network parameters of $K = 4$, $M = 8$, $N = 96$

SST-IE Nets	BN	PN	FBN
# of parameters	3258464	340336	407592
kernel size	(3,3)	(3,11,3)	–
padding	(1,1)	(1,5,1)	–
# of Sub-nets	$L_{\text{BF}} = 4$	$L_{\text{PS}} = 4$	$L_{\text{F}} = 4$
hidden features	$C_{\text{BF}} = 200$	$C_{\text{PS}} = 8$	$C_{\text{FBF}} = 500$
Loss weights	$w_1, w_2 = 5, 10$	$\gamma_1, \gamma_2 = 1, 1$	$\alpha_1 = 10$
For ($K = 4$, $M = 4$, $N = 16$), only C_{PS} differs, set to 24.			

two-stage approach invokes PhaseNet and BeamNet only once. In our design, the input to PN is based on the symbol-level cascaded channel $\mathbf{H}_k\mathbf{W}$, rather than the cascaded channel $\mathbf{H}_k$ used in [6], enabling natural integration into an iterative process. As illustrated in Fig. 4, each iteration applies PN, followed by BN and FBN.

Initially, the input to PN is $[\mathbf{H}_1\mathbf{W}_{ini}|\cdots|\mathbf{H}_K\mathbf{W}_{ini}]$. The output of PN at the t-th iteration, denoted as $\hat{\boldsymbol{\theta}}^{(t)}$, is then multiplied by the cascaded channel to produce $[\hat{\boldsymbol{\theta}}^{(t)}\mathbf{H}_1;\cdots;\hat{\boldsymbol{\theta}}^{(t)}\mathbf{H}_K]$, which serves as the input to BN. The output of BN, $\hat{\mathbf{W}}^{(t)}$, is then combined with the effective channel to form $[\hat{\boldsymbol{\theta}}^{(t)}\mathbf{H}_1\hat{\mathbf{W}}^{(t)};\cdots;\hat{\boldsymbol{\theta}}^{(t)}\mathbf{H}_K\hat{\mathbf{W}}^{(t)}]$, which is used as the input to FBN. The output of FBN, denoted as $\hat{\mathbf{W}}_{\text{F}}^{(t)}$, is combined with the cascaded channel to form $[\mathbf{H}_1\hat{\mathbf{W}}_{\text{F}}^{(t)}|\cdots|\mathbf{H}_K\hat{\mathbf{W}}_{\text{F}}^{(t)}]$, which serves as the input to PN in the $(t+1)$-th iteration. This iterative process continues, progressively enhancing the sum-rate performance through joint optimization of $\boldsymbol{\theta}$ and $\mathbf{W}$.

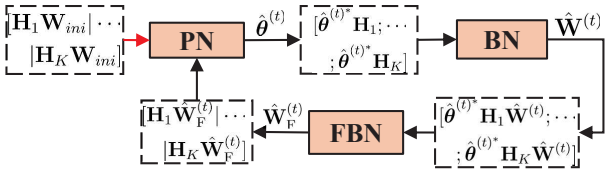


Fig. 4: Iterative execution.

D. Permutation enhancement for iterative execution

In addition, we adopt a permutation-based strategy to enhance the sum-rate performance of SST-IE. Given a randomly generated $\boldsymbol{\theta}_{rnd}$, we compute the initial beamforming matrix $\mathbf{W}_{ini}$ as defined in (3). We then generate all $K!$ permutations of the columns of $\mathbf{W}_{ini}$, resulting in $K!$ distinct initial power allocations across users. The iterative execution process is run for each permuted $\mathbf{W}_{ini}$, and the configuration yielding the highest sum-rate is selected. We observe that this permutation enhancement significantly improves SST-IE performance, while offering negligible gains for the BCD algorithm.

IV. Simulation Results

Fig. 1 illustrates the layout of the simulated multi-user network, where the BS and RIS are positioned at $(0\text{m}, 0\text{m})$ and $(200\text{m}, 10\text{m})$, respectively. We consider $K = 4$ users randomly distributed within a circular area centered at $(200\text{m}, 30\text{m})$ with a radius of 10 m. Two system configurations are evaluated: 1) $M = 8$ base station antennas and $N = 96$ RIS unit-cells; 2) $M = 4$ and $N = 16$. Assuming a bandwidth of $B_0 = 180$ kHz, the thermal noise power is computed as $-185 + 10\log_{10}(B_0) = -132.45$ dBm.

We generate 76544 channel realizations for training, each consisting of $\mathbf{G}$ and $\{\mathbf{h}_k\}_{k=1}^K$ as defined in the channel model in (1) of the paper. Channel generation follows the setup in [3] with user locations randomly selected within a 10-meter-radius circle (see Fig. 1). For each realization, $\mathbf{G}$ and $\{\mathbf{h}_k\}_{k=1}^K$ are multiplied to form the cascaded channels $\{\mathbf{H}_k\}_{k=1}^K$. To support supervised learning in the proposed SST-IE, we generate four BCD solution pairs $\{\mathbf{W}_{\text{BCD}}^{(j)}, \boldsymbol{\theta}_{\text{BCD}}^{(j)}\}$ per each sample, corresponding to iterations $j \in \{0, 25, 100, 500\}$. Each sample thus includes the cascaded channels and four associated BCD-generated beamforming and phase shift solutions. The batch size is set to 256, and to allow the network to learn across all four solution pairs, the effective batch size is 1024. We train the networks for 400 epochs using the Adam optimizer with an initial learning rate of 0.001.

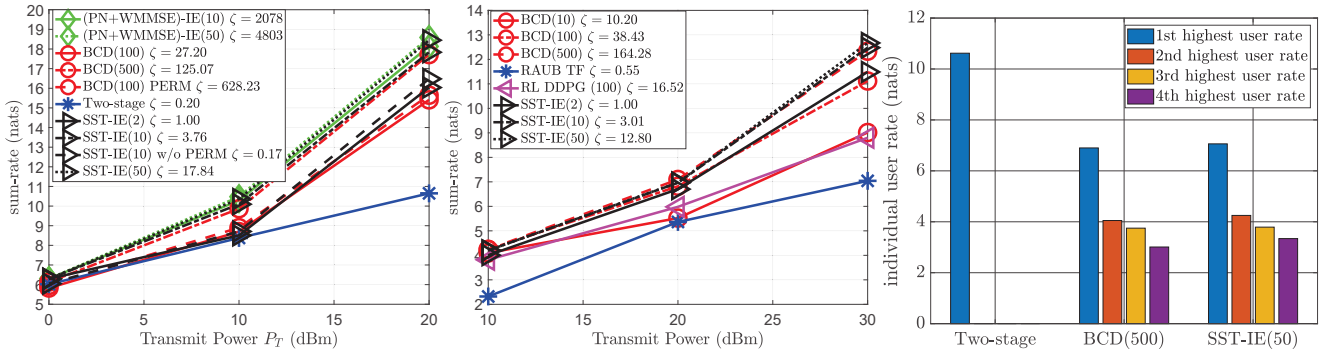
In Fig. 5a, we compare the performance of SST-IE with the conventional optimization-based BCD algorithm [3] and the neural network based two-stage method [6], averaged over 256 validation channel realizations, under the setting $K = 4$, $M = 8$, and $N = 96$. For fair comparison, the PhaseNet and BeamNet in the two-stage method use architectures similar to the PN and BN in SST-IE. The number in parentheses indicates the number of iterations: for example, BCD (500) and SST-IE (10) refer to BCD and SST-IE executed with 500 and 10 iterations, respectively. We use ζ to denote the ratio of the simulation time of a method to that of SST-IE (2).

Notably, the two-stage method, despite having the shortest simulation time ($\zeta = 0.20$), yields the worst sum-rate performance—especially in the range $P_T = 10 \sim 20$ dBm. In this regime, SST-IE (2) achieves a substantial gain (nearly 5.5 at $P_T = 20$ dBm) over the two-stage method, though with roughly 5× longer simulation time. SST-IE (2) also outperforms BCD (100), while being about 27× faster. Further improvement is seen with SST-IE (10), which gains around 1.8 at $P_T = 20$ dBm over SST-IE (2), at the cost of a 3.8× increase in simulation time. SST-IE (10) also slightly outperforms BCD (500). With 50 iterations, SST-IE achieves a notable sum-rate gain—exceeding BCD (500) by 0.7—while being about 7× faster. Although BCD (100) with permutation enhancement offers only a marginal gain over BCD (100), it incurs about 23× more simulation time. In contrast, permutation significantly benefits SST-IE (10), offering a gain of about 1.4 at $P_T = 20$ dBm. The main takeaway from Fig. 5a is that SST-IE (10) offers an effective trade-off between complexity and sum-rate in the high-SNR regime.

In Fig. 5a, we compare BN-FBN and the conventional optimization-based WMMSE algorithm [2] under the iterative execution framework by replacing the trained BN and FBN modules with the WMMSE algorithm, while keeping the trained PN module fixed. In this setting, WMMSE and PN iteratively refine both beamforming and phase shifts—referred to as (PN+WMMSE)-IE. The performance curves for (PN+WMMSE)-IE (10) and (50), corresponding to 10 and 50 iterations, are shown in green. While (PN+WMMSE)-IE (50) slightly outperforms SST-IE (50) in sum-rate, it incurs much higher inference time—approximately 269× slower than SST-IE (50). These results show that BN-FBN achieves comparable performance to WMMSE when paired with the same PN under

Authorized licensed use limited to: The University of Utah. Downloaded on July 25,2025 at 17:04:57 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Wireless Communications Letters. 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/LWC.2025.3584268



(a) $K = 4$, $N = 96$, and $M = 8$ (b) $K = 4$, $N = 16$, and $M = 4$, (c) Power allocation ($N = 96$, $M = 8$)

Fig. 5: Performance comparisons.

iterative execution.

In Fig. 5b, we compare the performance of SST-IE with BCD, RAUB-TF [9], and RL-DDPG [10] under a smaller system with $K = 4$, $N = 16$, and $M = 4$, averaged over 256 validation channel realizations. RAUB-TF is a two-stage unsupervised learning method based on a Transformer architecture, while RL-DDPG employs reinforcement learning via the Deep Deterministic Policy Gradient algorithm. Results show that both RAUB-TF and RL-DDPG significantly underperform compared to SST-IE (10). While RAUB-TF offers faster inference—about 5.5 times faster than SST-IE (10)—its sum-rate performance is substantially lower. RL-DDPG also yields inferior performance and suffers from approximately 5.5 times longer inference time compared to SST-IE(10). We note that the RL-DDPG (100) shown in Fig. 5b is obtained using 100 time-steps per channel realization. Both RAUB-TF and RL-DDPG tend to allocate most transmit power to a single user, limiting their sum-rate performance. In contrast, SST-IE and BCD promote more balanced power allocation across users, improving multi-user throughput. Notably, SST-IE(10) and SST-IE(50) outperform BCD(500) in sum-rate by approximately 0.2 and 0.3, respectively, while achieving 54.6× and 12.8× faster simulation times.

In Fig. 5c, we compare the performance of different algorithms in supporting multi-user transmission. For each channel realization, the users' individual rates are ranked from the highest to the lowest. The ranked rates are then averaged over 256 validation channel realizations to facilitate the comparison. At $P_T = 20$ dBm, we observe that the two-stage method supports only single-user transmission, with an average individual rate of 10.6 for the top-ranked user per channel realization, and 0 for the others. In contrast, both the BCD and SST-IE methods enable multi-user transmissions, with more balanced ranked individual rates across users. Specifically, SST-IE (50) offers not only a higher sum-rate than BCD (500), but also higher ranked individual rates for all four of them.

V. Conclusion and Future Work

In this work, we propose a novel SST-IE framework for joint optimization of BS transmit beamforming and RIS phase shifts in RIS-aided networks. Notably, SST-IE achieves sum-rate performance that surpasses the BCD algorithm using as few as 10 online iterations. By leveraging semi-supervised learning and separately training its core modules, SST-IE simplifies the training process while promoting fair rate allocation across users—a key improvement over existing two-stage neural network designs and the RL-DDPG approach, both of which tend to concentrate power on a single user. Moreover, SST-IE outperforms the RL-DDPG method with significantly lower training complexity. Future work will explore extending the semi-supervised learning strategy beyond BCD-labeled data and investigating alternative network architectures to further improve performance and reduce execution-phase iterations.

References

[1] 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.

[2] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[3] H. Guo, Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks," *IEEE transactions on wireless communications*, vol. 19, no. 5, pp. 3064–3076, 2020.

[4] Y. Zhao, F. Lin, K. Liu, and J. Zhang, "Ris-aided rate optimization research for ps-swipt system," in *2023 IEEE 11th International Conference on Information, Communication and Networks (ICICN)*. IEEE, 2023, pp. 149–154.

[5] Y. Wang, L. Fang, S. Cai, Z. Lian, Y. Su, and Z. Xie, "Low-complexity algorithm for maximizing the weighted sum-rate of intelligent reflecting surface-assisted wireless networks," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 10 490–10 499, 2023.

[6] H. Song, M. Zhang, J. Gao, and C. Zhong, "Unsupervised learning-based joint active and passive beamforming design for reconfigurable intelligent surfaces aided wireless networks," *IEEE communications letters*, vol. 25, no. 3, pp. 892–896, 2020.

[7] J. Cho, X. Huang, and R.-R. Chen, "Two time-scale learning for beamforming and phase shift design in RIS-aided networks," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 2627–2632.

[8] Y. Jiao, Y. Han, X. Li, and S. Jin, "Unsupervised learning-based joint precoding and phase shift design for RIS-assisted mmWave communication systems," in *2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2022, pp. 303–308.

[9] Y. Cui, G. Wang, D. Wu, P. He, R. Wang, and Y. Liu, "Ris-assisted unsupervised beamforming in internet of vehicles," *IEEE Transactions on Vehicular Technology*, 2024.

[10] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[11] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

Authorized licensed use limited to: The University of Utah. Downloaded on July 25,2025 at 17:04:57 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.